Knowledge-Based Systems 262 (2023) 110248

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems

Mohamed Abdel-Basset [a], Reda Mohamed [a], Mohammed Jameel [b,c], Mohamed Abouhawwash [b,d,*]

[a] Faculty of Computers and Informatics, Zagazig University, Shaibet an Nakareyah, Zagazig, 44519 Ash Sharqia Governorate, Egypt
[b] Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt
[c] Department of Mathematics, Faculty of Science, Sana'a University, Sana'a 13509, Yemen
[d] Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA

ARTICLE INFO

Article history:
Received 2 April 2022
Received in revised form 26 December 2022
Accepted 28 December 2022
Available online 2 January 2023

Keywords:
Optimization
Nature-inspired
Swarm algorithms
Nutcracker optimization algorithm
Constrained optimization

ABSTRACT

This work presents a novel nature-inspired metaheuristic called Nutcracker Optimization Algorithm (NOA) inspired by Clark's nutcrackers. The nutcrackers exhibit two distinct behaviors that occur at separate periods. The first behavior, which occurs during the summer and fall seasons, represents the nutcracker's search for seeds and subsequent storage in an appropriate cache. During the winter and spring seasons, another behavior based on the spatial memory strategy is regarded to search for the hidden caches marked at different angles using various objects or markers as reference points. If the nutcrackers cannot find the stored seeds, they will randomly explore the search space to find their food. NOA is herein proposed to mimic these various behaviors to present a new, robust metaheuristic algorithm with different local and global search operators, allowing it to solve various optimization problems with better outcomes. NOA is evaluated on twenty-three standard test functions, test suites of CEC-2014, CEC-2017, and CEC-2020 and five real-world engineering design problems. NOA is compared with three classes of existing optimization algorithms: (1) SMA, GBO, EO, RUN, AVOA, RFO, and GTO as recently-published algorithms, (2) SSA, WOA, and GWO as highly-cited algorithms, and (3) AL-SHADE, L-SHADE, LSHADE-cnEpSin, and LSHADE-SPACMA as highly-performing optimizers and winners of CEC competition. NOA was ranked first among all methods and demonstrated superior results when compared to LSHADE-cnEpSin and LSHADE-SPACMA as the best-performing optimizers and the winners of CEC-2017, and AL-SHADE and L-SHADE as the winners of CEC-2014.

Published by Elsevier B.V.

1. Introduction

During the past two decades, meta-heuristic (MH) algorithms have stirred great interest in the evolutionary computation community. This interest is due to the high precision, optimization speed, and low computational complexity of these algorithms. MH approaches have proven to be highly capable of solving complex optimization problems in various applied disciplines. Chamaani et al. [31] presented the application of these methods in the optimization of time and frequency domains for antennas. Penghui et al. [32] discussed the application of MH approaches in the development of a hybrid artificial intelligence model for soil temperature prediction. Gao et al. [33] proposed applying these approaches to dynamic routing problems. These algorithms also show efficiency in sequential data processing in medical applications, which is highly valuable for various gene modeling activities [34]. Furthermore, protein structure modeling and assignments can be solved by using specific derivatives of MH methods, as discussed in [35]. Analysis of the medical results can also depend on MH methods. In the field of medical diagnosis, the customized version of the MH approach has shown excellent results in analyzing medical data for classification [36,37] and radiotherapy planning [38]. Recently, MH approaches are reported to support text mining in real-time applications [39] and spectral clustering [21]. These methods in protein encoding are reported in medical applications [40]. In physical applications, many MH approaches have been devoted to extracting parameters of different photovoltaic models [41–45]. MH methods were also used in engineering for a variety of purposes. Mirghasemi et al. [46] used these methods to reduce image noise. Zhang et al. [47] offered particle filtering for robot localization. Das et al. [48] provided a detailed examination of these methods, including recent advancements, a range of conceivable applications, and theoretical

* Corresponding author at: Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA.
E-mail addresses: mohamedbasset@ieee.org (M. Abdel-Basset), redamoh@zu.edu.eg (R. Mohamed), moh.jameel@su.edu.ye (M. Jameel), abouhaww@msu.edu, saleh1284@mans.edu.eg (M. Abouhawwash).

https://doi.org/10.1016/j.knosys.2022.110248
0950-7051/Published by Elsevier B.V.

Table 1
A summary of Nature-Inspired Algorithms, for which the behavior of animals or plants was an inspiration for the optimization technique.

Method	Ref.	Inspiration from nature	Year
Particle Swarm Optimization (PSO)	[1]	Flock of fish, birds, etc.	1995
Ant Colony Optimization (ACO)	[2]	Ants in a colony	1999
Marriage in Honey Bees Optimization (MPO)	[3]	Honey bees	2001
Artificial Fish Swarm Algorithm (AFSA)	[4]	Fish swarm	2003
Termite Algorithm (TA)	[5]	Termites in a colony	2006
Artificial Bee Colony (ABC)	[6]	Bees in a colony	2007
Cuckoo Search Algorithm(CS)	[7]	Cuckoos	2009
Firefly algorithm (FA)	[8]	Fireflies	2010
Bat-Inspired Algorithm (BA)	[9]	Bats	2010
Flower Pollination Algorithm	[10]	Flower pollination	2012
Krill Herd (KH)	[11]	Krill	2012
Bird Mating Optimizer (BMO)	[12]	Birds	2013
Spider Monkey Optimization algorithm (SMO)	[13]	Spider Monkey	2014
Gray Wolf Optimizer (GWO)	[14]	Pack of wolves	2014
Ant Lion Optimizer (ALO)	[15]	Ant lions	2015
Crow search algorithm (CSA)	[16]	Crows	2016
Whale Optimization Algorithm (WOA)	[17]	Whale	2016
Salp Swarm Algorithm (SSA)	[18]	Salp	2017
Satin bowerbird optimizer (SBO)	[19]	bower bird	2017
Emperor penguin optimize (EPO)	[20]	Emperor penguin	2018
Squirrel Search Algorithm (SSA)	[21]	Squirrels	2018
Harris Hawks Optimization (HHO)	[22]	Hawks	2019
Seagull optimization algorithm (SOA)	[23]	Seagull	2019
Slime Mould Algorithm (SMA)	[24]	Mould	2020
Chimp optimization algorithm	[25]	Chimp	2020
Red fox optimization algorithm (RFO)	[26]	Red fox	2021
Artificial gorilla troops optimizer (GTO)	[27]	Gorilla	2021
Aquila Optimizer (AO)	[28]	Aquila	2021
Carnivorous plant algorithm (CPA)	[29]	Carnivorous plant	2021
Artificial hummingbird algorithm (AHA)	[30]	Hummingbirds	2022

foundations. MH approaches are expected to be greatly utilized in different areas of life due to their high accuracy and ease of implementation.

Nature is the source of inspiration for most MH methods. We can see amazing ways animals hunt, reproduce, or travel in search of food. We can also see exciting phenomena in the plant world and natural elements that inspire the development of science. These behaviors have recently been modeled in a variety of optimization techniques, the results of which are summarized in Table 1.

Among the state-of-the-art MH algorithms, we have many well-known classic models, such as simulated annealing, proposed in [49], where the concept of burning iron and other metals was used to construct one of the earliest optimization techniques. The genetic algorithm (GA), which is presented in Ref. [50], is one of the most popular optimization algorithms that simulate the evolution of genes. The GA comprises different types of randomness. This algorithm randomly performs selection, reproduction, mutation, and other operations, which greatly helps in avoiding local optima. Random behavior is the main feature of optimization algorithms, which leads to non-deterministic results. The idea of human imitation of nature is successful because the development of a model of interaction and communication of life between individual animals is selected as a strategy for improvement. Particle swarm optimization, which is presented in [1], is the first and most popular among these algorithms. Differentiable evolution, which was proposed in [51], is inspired by the concept of the natural phenomenon of evolution. The ant colony optimization proposed in [2] is inspired by the behavior of ants in foraging. The artificial bee colony presented in [6] simulates the behavior of bees in foraging and the production of honey. The cuckoo search algorithm proposed in [7] simulates the behavior of cuckoo breeding parasitism. Ref. [9] presented a simulation of the echolocation abilities of bats, wherein a custom echolocation-based locomotion model was proposed in Bat Algorithm (BA). Ref. [10] proposed a flower pollination algorithm, which is inspired by the pollination process of flowers. Ref. [15] presented an ant lion optimizer, which simulates the manner the ant lion ambushes to search in the solution space for the optimal point. The salp swarm algorithm (SSA) presented in [18] simulates the swarming behavior of salps when navigating and foraging in oceans. Many other optimization algorithms based on certain characteristics of animal life, such as unique hunting method, migratory method in search of food or breeding, and other lifestyles, have been proposed in recent years. The seagull optimization algorithm presented in [23] simulates the migration behaviors of a seagull in nature. Ref. [24] proposed the slime mould algorithm, which simulates the oscillation style of slime mould in nature. Ref. [26] presented a red fox optimization algorithm, which simulates the strange shapes of the red fox during a hunt because it uses different strategies to distract the prey, approach it, and then pounce on it. The carnivorous plant algorithm presented in [29] is one of the recently introduced optimization algorithms that simulates how carnivores hunt to survive in a harsh environment. The latest algorithm is inspired by hummingbirds, which mimics hummingbirds' skills in flight and the intelligent strategy of these birds in foraging [30]. The chimp optimization algorithm (ChOA) is improved in [52] to help search agents make the transition from exploratory to exploitative mode. Furthermore, in [53], the niching technique is integrated with the ChOA to prevent premature convergence by preserving the chimp's diversity during the optimization process. Ref. [54] presented a new variant of the gray wolf optimizer, dubbed the cross-dimensional coordination gray wolf optimizer (CDCGWO), which employs a novel learning technique to preserve the wolf's diversity for avoiding premature convergence in multimodal optimization tasks.

MH methods are called stochastic methods. As the name of these techniques suggests, they initially randomly perform optimization. The optimization process begins by creating a set of random solutions. These initial solutions are combined, transformed, or developed via iterations or generations through a

predetermined number of steps. These methods are different from each other because they use distinct ways to combine, transfer, or develop solutions during improvement.

MH methods have become a powerful alternative to deterministic methods for solving various optimization problems. Although deterministic methods are effective in solving problems involving linear (unimodal) search spaces, they tend to get stuck at local optima when applied to nonlinear search space problems, including non-convex real-world problems [55]. Moreover, deterministic algorithms require derivative information and mechanically and iteratively work without any randomness. By contrast, MH methods do not need derived information and work within a random nature that allows them to search for all optimal solutions in the search space while avoiding local optima. MH optimization methods have two key features: (1) exploration and (2) exploitation. Exploration is the ability to search space globally. This ability is linked to escaping local optima and preventing stagnation in the same. Meanwhile, exploitation is the ability to locally explore promising regions to improve their quality. A good compromise between these two properties results in optimal performance. These properties are used by all population-based algorithms, but with various operators and processes.

Despite the success of traditional and modern algorithms, no algorithm guarantees to find the global optimal solution for all optimization problems. This notion is logically proven by "No Free-lunch" [56] theory. Accordingly, researchers are motivated to design new and more efficient algorithms. In this article, we present a new optimization algorithm inspired by the behavior of the nutcracker in the search for good quality food and its behavior in the storage of food. This work also presents the unique behavior of birds in search of stored food again and retrieve it. The strange and unique behavior of the aforementioned nutcracker prompted researchers to design a new optimization algorithm model, which proved effective in solving various optimization problems.

The remainder of this paper is structured as follows: Section 2 presents a brief overview of Cark's nutcrackers in nature. Section 3 details our proposed NOA. Section 4 outlines the implementation method for the proposed algorithm. Section 5 presents the optimization outcomes and analyses for many test mathematical problems. Some real-world optimization problems and their outcome are presented in Section 6. Finally, Section 7 provides the conclusion and future work.

2. Cark's nutcrackers in nature

Clark's nutcrackers, *Nucifraga columbiana*, are a high latitude resident corvid species (Family Corvidae) that live in the mountain regions of the western USA and Canada [57,58]. Like other species of the Corvidae family, Clark's nutcrackers are distinguished by their extraordinary intelligence. Clark's nutcrackers are pale gray birds with black wings (see Fig. 1). These birds often live alone, unlike other species of the Corvidae family that live in family groups, such as crows. The popularity of these birds stems from the positive relationship that they have with pine trees. Pine seeds are the primary food source for Clark's nutcrackers. Meanwhile, Clark's nutcracker is the primary seed disperser of whitebark pine (*Pinus albicaulis*) [59,60]. Several studies have shown that the declining white pine numbers negatively affect nutcracker numbers [61–64].

Clark's nutcrackers (hereafter referred to as the "nutcracker") are regarded as one of the most specialized scatter-hoarding birds and are an effective distributor of whitebark pine seeds [65]. The nutcracker's unique method of transporting and storing pine seeds is unique. A single nutcracker caches about 22,000 to 33,000 pine seeds, distributed over thousands of separate positions in autumn with a good seed crop [58,66]. Nutcrackers

Fig. 1. Nutcracker bird.

are dependent on stored seeds in winter, which they can extract even through the deep snow cover. Caching continues until the seed crop is depleted or inclement weather intervenes. Moreover, nutcrackers collect seeds from trees with high efficiency. Nutcrackers can also distinguish between edible and aborted seeds; they can select cones that have an above-average number of good seeds and seem to concentrate on trees that produce cones with many good seeds. A pinecone is an organ of the pine tree containing its reproductive structures (seeds). The nutcracker has a sublingual bag that can carry up to 90 pinon pine seeds per trip [67–69]. The seeds can be transported up to 32 km from the collection area (pine trees) to temporary storage positions where they are buried in groups (usually 4–5 pine seeds) in underground caches at a depth of 2–3 cm [67,70–73]. The seeds are buried on treeless uplands and on south-facing slopes where the snow melts earliest in the spring [74,75].

Nutcrackers choose high-quality pine seeds because they are large and easily harvested [75]. Additionally, nutcrackers prefer locally abundant trees because they collect seeds from cones on the tree with a higher cone density [76,77]. Nutcrackers also choose viable seeds, which reflect positively on the pine trees because these increase their chances of reproduction. Moreover, nutcrackers store considerable seeds in temporary storage, reducing competition for moisture and space [72,78]. The depth at which the seeds are stored is compatible with germination requirements, and many positions selected appear suitable for seed germination [61,64,78,79].

The nutcracker uses spatial memory to retrieve food stored in the field [80]. Several studies show that nutcrackers can remember storage locations for up to 9 months [80–82]. Nutcrackers use visual cues to recall the exact position of each cache [83–85]. Researchers in this field have found that nutcrackers use two or more objects as cues [58,83,86]. When a nutcracker digs its hole, it will see two or three things attached to the site: an uneven rock, a bush, the trunk of a tree, and so on. Studies show that nutcrackers can remember up to 80% of buried stores. However, nutcrackers sometimes do not get their food inside the storage for various reasons; either the storage was stolen by another nutcracker/bird, or it was destroyed due to nature or other unknown factors.

In this work, a novel nature-inspired MH algorithm was introduced based on the search, cache, and recovery behaviors of the nutcracker. In the next section, these behaviors are mathematically modeled and a nutcracker optimization algorithm (NOA) is proposed.

3. Nutcracker optimization algorithm (NOA)

A bio-inspired optimization algorithm (NOA) based on the nutcracker's intelligent behaviors is presented in this section.

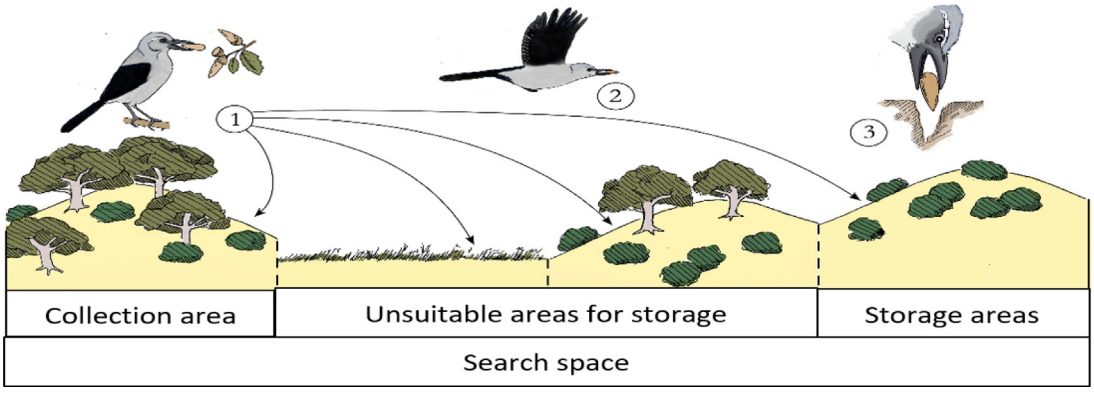


Fig. 2. Search space in NOA.

As previously mentioned, the behavior of the nutcracker can be divided into two main parts: the first one is to collect pine seeds (food) and store them; and the second one is to search for and retrieve storage places. These two behaviors are characterized by their occurrence in two different periods. The first behavior occurs in the summer and fall periods. Meanwhile, the second behavior occurs in the winter and spring periods. In the proposed algorithm, we simulate the behavior of a nutcracker based on the aforementioned two main behaviors. The two main strategies are (i) foraging and storage strategy; and (ii) cache-search and recovery strategy. These strategies can be explained as follows.

3.1. Foraging and storage strategy

Nutcrackers start randomly searching for pine seeds (a food source) in the pine forest. Hereinafter, we will call the pine forests a collection area. Nutcrackers select pine cones that contain good seeds. If nutcrackers do not find the desired seed in the selected pine tree, then they will explore good seeds in other pine trees in the collection area. This behavior is called the first exploration phase. Then, nutcrackers exploit the abundance of seeds and store them away from the collection area. They transport the forage to suitable storage sites, usually high areas without heavy afforestation, as shown in Fig. 2. Low and wooded sites are vulnerable to poaching or damage by rodents and birds, in addition to other weather-related reasons. This behavior is called the first exploitation phase.

3.2. Cache-search and recovery strategy

During winter, nutcrackers begin to explore the stored seeds. This behavior is called the second exploration phase. Nutcrackers do not only randomly search for their caches but also use objects near the caches as clues to determine their location. Furthermore, nutcrackers use spatial memory as the main mechanism to help recover their caches. They use multiple objects near a single cache and have the ability to remember the location of the cache for a long time. Nutcrackers exploit food caches to feed themselves and their young for up to 6 months. This behavior is called the second exploitation phase. Nutcrackers rely more on spatial memory to survive. However, nutcrackers cannot sometimes find the cache, so they seek another cache to recover their food.

In reality, both of the two types behaviors discussed above occur at two distinct times: nutcrackers store pine seeds in late summer and fall every year and buried seeds, which will serve as their main food source, during winter and spring. All the above-mentioned nutcrackers' behaviors will be mathematically simulated to propose a novel MH approach for solving optimization problems.

3.3. Framework of the proposed NOA

The overall goal of this work is to present a new optimization method that mimics the behavior of nutcrackers in foraging, storing, and recovering pine seeds. The following principles derived from this behavior achieve the basic assumptions of this algorithm:

- A nutcracker lives in individual form.
- A nutcracker has two behaviors in two separate periods.
- A nutcracker seeks efficient pine seeds in the collection area.
- A nutcracker carries pine seeds to places far from the collection area and buries them.
- A nutcracker memorizes the location of the buried seeds.

In reality, nutcrackers can have multiple caches. However, we assume that each nutcracker only has one food source, one cache, for simplicity. Specifically, a solution X_i is equivalent to a nutcracker and/or a food source. In future studies, we can easily extend to multiple caches for each nutcracker and multiple nutcrackers for multi-objective optimization problems.

Fig. 3 illustrates the framework of the proposed NOA. NOA has a very simple structure consisting of two main strategies representing two different behaviors that occurred during two separate periods. The first strategy represents the nutcracker's behavior in searching for forage and then storing it in an appropriate cache. By contrast, the second strategy represents the behavior of the nutcracker in searching for and retrieving stored places. In NOA, the search space can be divided into two areas: the first one is the collection area, which means the area from which the nutcracker collects its food; the second one is a storage area, which means the area in which the nutcracker stores its food. The proposed information-sharing mechanism between the two strategies in NOA is also shown in Fig. 3, where each strategy consists of local exploitation and global exploration. In summary, this figure depicts the behavior of nutcrackers mimicked by NOA in order to present a novel metaheuristic algorithm capable of tackling a variety of optimization problems. At the start of the optimization process, a group of the nutcrackers will go to explore the search space for finding their food, represented in the pine seeds. The discovered pine seeds will be stored in the appropriate caches to be retrieved through the winter and spring seasons until the nutcrackers reproduce and survive. The first mechanism followed by the nutcrackers through the summer and fall periods is defined as the foraging and storage strategy, which employs an exploration operator as the first stage to explore the search space for finding the most promising regions, which include the pine seeds. Following that stage, the exploitation operator will be fired to store the discovered pine seeds in an appropriate cache

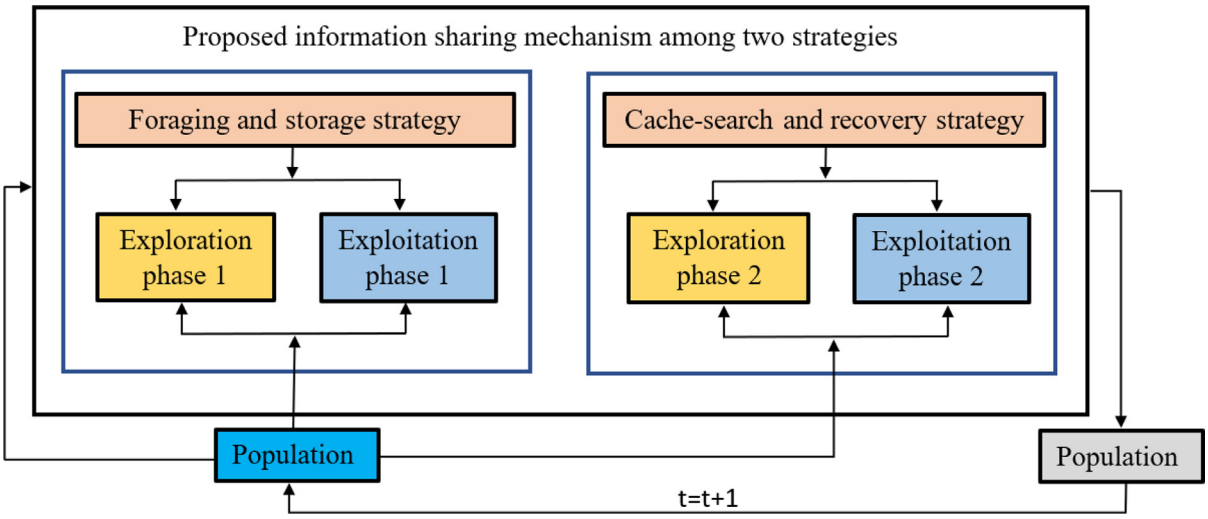


Fig. 3. The framework of the proposed NOA.

marked with various objects or markers as a cue to help them find them through the winter and spring seasons. Through the winter and spring seasons, a different mechanism based on the spatial memory strategy is followed by the nutcracker to search for and retrieve the stored pine seeds. This mechanism is referred to as the cache-search and recovery strategy. In this strategy, an exploration operator based on the reference points that mark the hiding places of caches is employed to explore the regions around those reference points to find the hidden caches. When finding the hidden caches, the exploitation operator will be fired to retrieve the buried pine seeds. The detailed explanation for these two strategies is presented within the next sections.

3.3.1. Foraging and storage strategy

The above-mentioned mechanism can be divided into two main stages, namely, foraging and storage, which are described as follows:

• Foraging stage: Exploration phase 1

At this stage, the nutcrackers begin to take their initial positions/food positions, generated by Eq. (19), in the search space (collection area). Each nutcracker begins by checking the cone that contains the seeds in the initial position. If the nutcracker finds good seeds, then it will take them to the storage area to bury them in a cache. If the nutcracker cannot find good seeds, then it will seek another cone in another position within pine trees or other trees. This behavior can be mathematically modeled using the position update strategy as follows:

$$\vec{X}_i^{t+1} = \begin{cases} X_{i,j}^t & if\ \tau_1 < \tau_2 \\ \begin{cases} X_{m,j}^t + \gamma \cdot \left(X_{A,j}^t - X_{B,j}^t\right) \\ \quad +\mu . \left(r^2 . U_j - L_j\right), & if\ t \le T_{max}/2.0 \\ X_{C,j}^t + \mu \cdot \left(X_{A,j}^t - X_{B,j}^t\right) \\ \quad +\mu . \left(r_1 < \delta\right) . \left(r^2 . U_j - L_j\right), & Otherwise \end{cases} & otherwise \end{cases} \tag{1}$$

where X_i^{t+1} is the new position of the ith nutcracker in the current generation t; $X_{i,j}^t$ is the jth position of the ith nutcracker in the current generation; U_j and L_j are vectors, including the upper and lower bound of the jth dimension in the optimization problem; γ is a random number generated according to the levy flight; $X_{best,j}^t$ is the jth dimension of the best solution obtained even now; A, C, and B are three different indices randomly selected from the population to facilitate exploration of a high-quality food source; τ_1, τ_2, r, and r_1 are random real numbers in the range of [0,1]; $X_{m,j}^t$ is the mean of the jth dimensions of all solutions of the current population in the iteration t; and μ is a number generated based on the normal distribution (τ_4), levy-flight (τ_5), and randomly between zero and one (τ_3) as shown in the following equation:

$$\mu = \begin{cases} \tau_3 & if\ r_1 < r_2 \\ \tau_4 & if\ r_2 < r_3 \\ \tau_5 & if\ r_1 < r_3 \end{cases} \tag{2}$$

where r_2 and r_3 are random real numbers in the range of [0,1]. Eq. (1) was suggested to investigate high-quality food sources. The first state of Eq. (1) simulates the probability that the nutcrackers discover good seeds on the first try. This notion means that the nutcrackers will not change their original jth dimension in the solution $\vec{X}_i^t$. The second state of Eq. (1) was proposed to allow the nutcrackers to globally explore random positions in the search space to enhance the search capability of NOA for exploring the search space as must as possible to avoid being stuck into local minima and reaching the promising regions that might include the near-optimal solution. The third state of Eq. (1) was proposed to allow the nutcrackers to explore positions around a solution randomly selected from the population in addition to moving globally in the search space with a probability δ to cover intractable regions by the second state of Eq. (1). The sensitivity analysis for estimating the best value for δ is introduced later. A and B were suggested to provide the nutcrackers with information about the new food position. τ_1 and τ_2 were suggested as scaling factors to manage the exploration ability in NOA. The proposed algorithm can alternate between local and global searches with a range of small and large values for τ_1 and τ_2. μ provides nutcracker information about the required direction with various step sizes to explore a new position. This exploration phase is depicted in Fig. 4(a) for four independent runs to show how far it is capable of exploring the regions within the search space. From this figure, it is clear that most of the search space is coverage by the solutions generated under the foraging stage through the whole optimization process to affirm that this stage is as effective as approved in the practical experiments conducted later.

• Storage stage: Exploitation phase 1

Nutcrackers begin by transporting the food obtained in the previous phase, exploration phase 1, to temporary storage locations (storage area). This stage is known as "exploitation phase 1", where the nutcrackers exploit pine seed crops and store them. Such behavior can be mathematically expressed as follows:

$$\vec{X}_i^{t+1(new)} = \begin{cases} \vec{X}_i^t + \mu \cdot \left(\vec{X}_{best}^t - \vec{X}_i^t\right) \cdot |\lambda| + r_1 \cdot \left(\vec{X}_A^t - \vec{X}_B^t\right) & if\ \tau_1 < \tau_2 \\ \vec{X}_{best}^t + \mu \cdot \left(\vec{X}_A^t - \vec{X}_B^t\right) & if\ \tau_1 < \tau_3 \\ \vec{X}_{best}^t \cdot l & Otherwise \end{cases} \tag{3}$$

where $\vec{X}_i^{t+1(new)}$ is a new position in the storage area of the nutcrackers in current iteration t, λ is a number generated according to the lévy flight, and τ_3 is a random number between 0 and 1. l is a factor that linearly decreased from 1 to 0 to diversify in the exploitation behavior of NOA. This variety in the exploitation operator of NOA will help in accelerating its convergence speed, in addition to avoiding stuck into local minima that might occur when searching in one direction.

The exchange between the foraging stage and the cache is applied according to the following formula to maintain the balance between exploration and exploitation operators through the optimization process:

$$\vec{X}_i^{t+1} = \begin{cases} \text{Eq. (1)}, & if\ \varphi > P_{a_1} \\ \text{Eq. (3)}, & otherwise \end{cases} \tag{4}$$

where φ is a random number between zero and one, and P_{a_1} represents a probability value that is linearly decreased from one to zero based on the current generation. Fig. 4(b) is presented to show the direction of the solutions generated using this stage, which affirms that this stage can steer the majority of the solutions in the direction of one region to cover the solutions there as much as possible while searching for the near-optimal solution. The flowchart of the exploration and exploitation processes in the first proposed strategy is depicted in Fig. 5 and listed in Algorithm 1.

Algorithm 1 Foraging and storage strategy

Output : $\vec{X}_{best}$
1. Initialize N nutcrackers using Eq.(19)
2. Evaluate each X_i and finding the one with the best fitness in $\vec{X}_{best}$
3. $t = 1$; //the current iteration
4. **while** $(t < T)$
5. φ is a random number between 0 and 1.
6. **for** i=1:N
7. **for** j= 1:d
8. **if** $\varphi > P_{a_1}$ /***Exploration phase1*/**
9. Updating $\vec{X}_i^{t+1}$ using Eq. (1) and Eq. (20)
10. **Else** /***Exploitation phase1*/**
11. Updating $\vec{X}_i^{t+1}$ using Eq. (3) and Eq. (20)
12. **End if**
13. **End for**
14. Update the current iteration t by $t = t + 1$
15. **End for**
16. **End while**

3.3.2. Reference memory

After storing the seeds in the cache, the nutcrackers activate the spatial memory. The nutcrackers take a picture in their mind to find the cache using specific objects as a cue to help remember their cache. These nutcrackers rely almost solely on their memory of where those caches are located to survive through winter. Nutcrackers carefully examine and draw the storage location. Hidden food is their only chance to reproduce and survive. Moreover, nutcrackers seem to take large terrain features or tall trees as signs, while local features are not useful because they are covered in snow in winter. Next, we will simulate the behavior of nutcrackers in retrieving their cache.

3.3.3. Cache-search and recovery strategy

This strategy can be divided into two main stages: Cache-search and recovery stages, which are described in detail within the next two subsections:

• Cache-search stage: Exploration phase 2

When winter comes, the trees are bare, and it is time to switch from hiding mode to exploration and search mode. The nutcrackers begin in search of their caches. We call this phase the second exploration. The nutcrackers use a spatial memory strategy to locate their caches. Nutcrackers most likely use multiple objects as signals for a single cache. We will assume that each cache has only two objects for simplicity. The objects, or markers, will be at different angles from the hiding place. In this stage, the nutcrackers begin to take their initial positions/cache positions, generated by Eq. (19), in the search space (storage area). We defined these objects as Reference Points (RPs). In NOA, two RPs of each cache/nutcracker of the population can be defined using the following matrix:

$$\text{RPs} = \begin{bmatrix} \overrightarrow{RP}_{1,1}^t & \overrightarrow{RP}_{1,2}^t \\ \vdots & \vdots \\ \overrightarrow{RP}_{i,1}^t & \overrightarrow{RP}_{i,2}^t \\ \vdots & \vdots \\ \overrightarrow{RP}_{N,1}^t & \overrightarrow{RP}_{N,1}^t \\ \vdots & \vdots \end{bmatrix} \tag{5}$$

where $\overrightarrow{RP}_{i,1}^t$ and $\overrightarrow{RP}_{i,2}^t$ represent RPs (objects) of the cache position $\vec{X}_i^t$ of the ith nutcracker in the current generation t. To illustrate the relationship between the following three elements: the nutcracker, cache, and the reference point, we assume that the three elements represent the vertices of a triangle. Nutcrackers can be located in different locations with various angles of view. Fig. 6 depicts three different locations for the nutcracker in 3D space, with three different locations of RP for one cache. We assume that the origin point, 3D space, is the location of the cache. The angle-of-view of nutcrackers varies. The first location is at an acute angle, the second location is at an obtuse angle, and the third location is at a right angle. A nutcracker can be found anywhere in 3D space, on the coordinate axes or the diagonal axes. The same is true for RPs.

Nutcrackers can recover hidden caches with high accuracy. However, according to relevant studies, approximately 20% of nutcracker attempts fail on the first try [83]. If the nutcracker cannot locate the food stored by the first RP, then it will identify it by the second RP. If the nutcracker cannot retrieve its cache a second time, then it will use the third reference. Two various equations are designed to generate the first and second RPs to improve the nutcracker exploration operation during searching for the hidden caches. The first RP is generated by updating the current position within the neighboring regions to find hidden caches around the nutcrackers. The mathematical formula for generating the first RP is as follows:

$$\overrightarrow{RP}_{i,k}^t = \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot \left(\left(\vec{X}_A^t - \vec{X}_B^t\right)\right), k = 1 \tag{6}$$

The second RP is generated by updating the current solution within the search space of the problem to help the nutcrackers

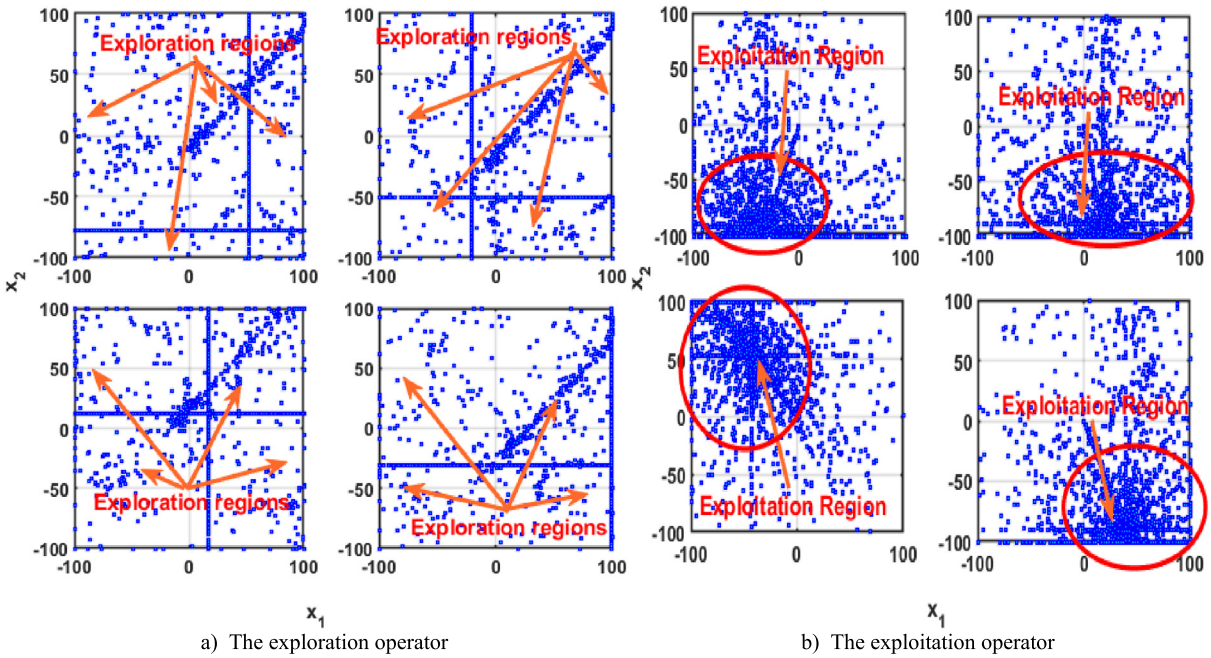


Fig. 4. Simulation of the exploration and exploitation operators in the Foraging and storage strategy.

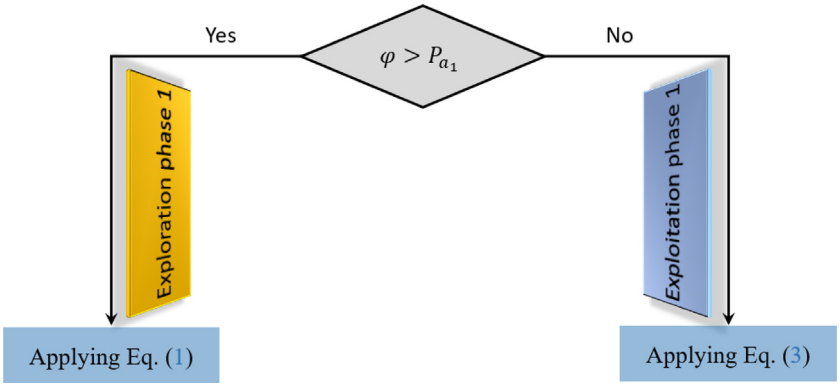


Fig. 5. Flowchart of the exploration and exploitation process in the first proposed strategy.

in exploring different regions for finding the hidden caches. The second RP is computed according to the following formula:

$$\overrightarrow{RP}_{i,k}^{t} = \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot \left(\left(\vec{U} - \vec{L}\right) \cdot \tau_3 + \vec{L}\right) \cdot \overrightarrow{U_2}, k = 2 \tag{7}$$

$$\overrightarrow{U_1} = \begin{cases} 1 & \overrightarrow{r_2} < P_{rp} \\ 0 & otherwise \end{cases} \tag{8}$$

where k denotes the RP index; $\overrightarrow{RP}_{i,k}^{t}$ is the RP k of the cache position $\vec{X}_i^t$ of the ith nutcracker in the current iteration t; $\vec{X}_i^t$ is the cache of the ith nutcracker in the current iteration t; $\vec{U}$ and $\vec{L}$ are the upper and lower boundaries for a D-dimensional problem, respectively; α linearly decreases from one to zero; $\overrightarrow{r_2}$ is a vector that includes values randomly generated between zero and one; $\vec{X}_A^t$ is the cache position of the Ath nutcracker in the current iteration t; τ_3 is a random number of the second RP in the range [0, 1]; θ is a random in the range [0, π]; and P_{rp} is a probability employed to determine the percentage of globally exploring other regions within the search space.

Eqs. (6) and (7) are used to generate two RPs for each nutcracker to recover its cache. When the RP is close to the cache, it can easily retrieve it. We assume that the nutcrackers do not have enough experience to select the appropriate RPs (i.e., near the cache) in the first generation. However, nutcrackers gain more experience in the storing process with time. Nutcrackers visualize landmarks near their warehouses so they can easily retrieve them later. α in these equations was proposed to train the nutcrackers and give them enough experience to choose a suitable site for RPs. In NOA, α prevents the early convergence of RPs toward the solution location (cache site). θ is the angle-of-view of the nutcracker and is chosen at random from 0 to π. Fig. 7 shows the different viewing angles of the nutcracker: (a) $\theta = 0$; (b) $0 < \theta < \pi/2$; (c) $\theta = \pi/2$; (d) $0 < \theta < \pi$; (e) $\theta = \pi$. When $\theta = \pi/2$, $\overrightarrow{RP}_{i,k}^{t} = \vec{X}_i^t$, indicating that the RP position lies on the same cache position. To address this issue (RP), Eqs. (6) and (7)

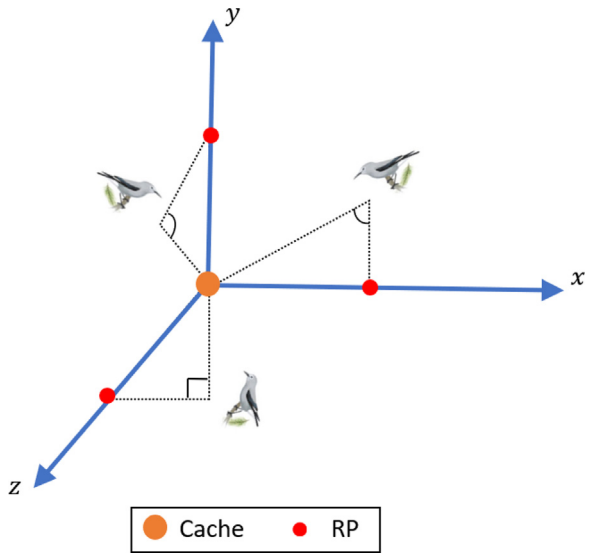


Fig. 6. Three different locations for the nutcracker in 3-D space, with three different locations of RP for one cache.

can be reformulated to the following equation:

$$\overrightarrow{RP}_{i,1}^{t} = \begin{cases} \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot \left(\left(\vec{X}_A^t - \vec{X}_B^t\right)\right) + \alpha \cdot RP, & if\ \theta = \pi/2 \\ \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot \left(\left(\vec{X}_A^t - \vec{X}_B^t\right)\right), & otherwise \end{cases} \tag{9}$$

and

$$\overrightarrow{RP}_{i,2}^{t} = \begin{cases} \vec{X}_i^t + \left(\alpha \cdot cos(\theta) \cdot \left(\left(\vec{U} - \vec{L}\right) \cdot \tau_3 + \vec{L}\right) + \alpha \cdot RP\right) \cdot \vec{U_2}, & if\ \theta = \pi/2 \\ \vec{X}_i^t + \alpha \cdot cos(\theta) \cdot \left(\left(\vec{U} - \vec{L}\right) \cdot \tau_3 + \vec{L}\right) \cdot \vec{U_2}, & otherwise \end{cases} \tag{10}$$

where $\overrightarrow{RP}_{i,1}^{t}$ represents the ith row, 1st column, in the matrix given in Eq. (5); $\overrightarrow{RP}_{i,2}^{t}$ represents the ith row, 2nd column, in the matrix given in Eq. (5). Each row index in the matrix represents the cache/nutcracker index. Meanwhile, every column index in the matrix represents the RP index. *RP* is a random position. The third term of the first state of Eqs. (9) and (10) (i.e., $\theta = \pi/2$), has been added to avoid the early convergence of RPs toward a possible solution. α ensures that the NOA converges on a regular basis, allowing the nutcracker to improve its RP selection in the next generations. α can be calculated according to the following equation:

$$\alpha = \begin{cases} \left(1 - \frac{t}{T_{max}}\right)^{2\frac{t}{T_{max}}}, & if\ r_1 > r_2 \\ \left(\frac{t}{T_{max}}\right)^{\frac{2}{t}}, & otherwise \end{cases} \tag{11}$$

where t and T_{max} indicate the current and maximum generations, respectively. The first state in Eq. (11) linearly decreases with the iteration to improve the convergence speed of the proposed algorithm. Meanwhile, the second state linearly increases to avoid being stuck into local minima, which might occur because of the first state.

During the optimization process, all nutcrackers gain enough experience in selecting the appropriate RPs. These nutcrackers update the storage locations/solutions according to the appropriate RPs. In NOA, all nutcrackers will apply the exploration mechanism to search for the most promising areas that might contain a near-optimal solution. With each generation passed, the algorithm will explore and exploit areas around caches with appropriate RPs to avoid getting stuck in local minima. The new position of a nutcracker can be updated using the following equation:

$$\vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^t, & if\ f\left(\vec{X}_i^t\right) < if\ (\overrightarrow{RP}_{i,1}^t) \\ \overrightarrow{RP}_{i,1}^t, & otherwise \end{cases} \tag{12}$$

where $\vec{X}_i^{t+1}$ is the new position/new cache of the ith nutcracker at iteration (t+1), $\vec{X}_i^t$ is the current position/current cache of the ith nutcracker in the current iteration t, and $\overrightarrow{RP}_{i,1}^t$ is the first RP of the current cache of the ith nutcracker at iteration t. Eq. (12) was proposed to guide NOA to explore and exploit the promising regions of the $\overrightarrow{RP}_{i,1}^t$. If NOA does not obtain the most promising regions around $\overrightarrow{RP}_{i,1}^t$, then it will explore them in other regions, such as the areas around the reference $\overrightarrow{RP}_{i,2}^t$, which is discussed below.

- **Recovery stage: Exploitation phase 2**

Fig. 8 depicts the possibilities that a nutcracker might encounter when searching for its cache. The first major possibility is that a nutcracker can remember the location of his cache using the first RP. The two possibilities shown in Fig. 8 are as follows: the first one is that the food exists, and the second is that the food does not exist. This behavior can be mathematically modeled according to the following equation

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t, & if\ \tau_3 < \tau_4 \\ X_{i,j}^t + r_1 \cdot \left(X_{best,j}^t - X_{i,j}^t\right) + r_2 \cdot \left(\overrightarrow{RP}_{i,1}^t - X_{C,j}^t\right), & otherwise \end{cases} \tag{13}$$

where $\vec{X}_i^{t+1}$ is the new position/cache of ith nutcracker at iteration (t+1), X_{ij}^t is the current dimension/cache of the ith nutcracker in the current iteration t, $\vec{X}_{best}^t$ is the best position/cache in iteration t, $\overrightarrow{RP}_{i,1}^t$ is the first RP of the current position/cache of the *ith* nutcracker in the current iteration t; r_1, r_2, τ_3 and τ_4 are random numbers created between zero and one, and C is the index of a solution selected randomly from the population.

The first state of Eq. (13) simulates the first possibility (food exists). This notion means that some dimensions in this cache/solution will have a chance to survive for the next generation. In reality, a cache that keeps pine seeds is good, so the nutcracker will prioritize food storage next time. The second state of Eq. (13) simulates the second possibility (food does not exist). In this case, the solution region is not promising, so the algorithm will apply an escape mechanism to avoid being stuck into local minima. Food from the cache is lost due to many reasons: either another nutcracker/bird stole the cache, or it was destroyed by certain natural factors, such as rain and snow. Eq. (13) state (1) allows a nutcracker to exploit promising regions in the search space to enhance the local search capability of NOA. Eq. (13) state (2) allows a nutcracker to explore random locations in the search space to enhance the global search capability of NOA.

The second major possibility is that the nutcracker does not remember the location of his hidden food using the first RP, so he will search for it using the second RP. In reality, a nutcracker keeps many RPs in his memory that he will take during early storage. Nutcrackers recover caches on the first try (with the first reference), but the probability of failing on the first try is considered in the proposed algorithm. The nutcrackers that cannot find their cache use local landmarks that might disappear when the weather changes between autumn (storage time) and

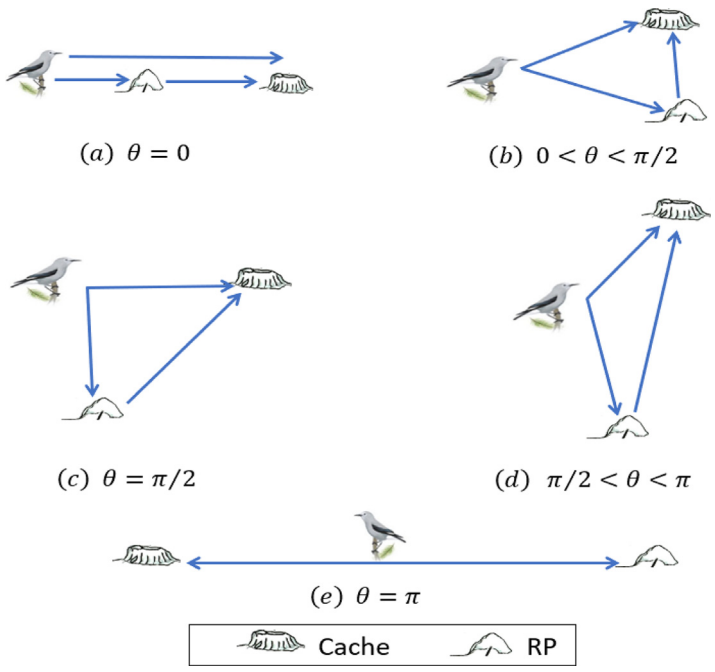


Fig. 7. Different viewing angles of the nutcracker for both cache and RP.

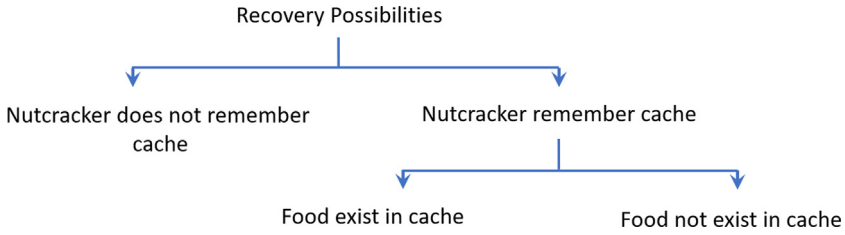


Fig. 8. Probable options for the Nutcracker to recover its cache.

winter (recovery time). The spatial memory of the nutcracker, Eq. (12), is updated using the second RP:

$$\vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^t, & if\ f\left(\vec{X}_i^t\right) < if\ (\overrightarrow{RP}_{i,2}^t) \\ \overrightarrow{RP}_{i,2}^t, & otherwise \end{cases} \tag{14}$$

where $\vec{X}_i^t$ is the current position/current cache of the ith nutcracker in the current iteration t, and $\overrightarrow{RP}_{i,2}^t$ is the second RP of the current cache of the ith nutcracker at iteration t. Eq. (14) offers an opportunity for the NOA to explore new regions around the second RP and exploit promising areas where a potential solution could be found. In NOA, a nutcracker is assumed to find its cache using the second RP, so Eq. (13) is updated based on the second RP:

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t, & if\ \tau_5 < \tau_6 \\ X_{ij}^t + r_1 \cdot \left(X_{best,j}^t - X_{ij}^t\right) + r_2 \cdot \left(\overrightarrow{RP}_{i,2}^t - X_{Cj}^t\right), & otherwise \end{cases} \tag{15}$$

where r_1, r_2, τ_5, and τ_6 are random numbers in interval [0,1]. The first state of Eq. (15) allows the algorithm to intensify the local search around the most appropriate areas that involve the near-optimal solution. In the contract, the second state of Eq. (15) allows the algorithm to search for new areas in the search space to enhance its global search ability. In summary, the simulation of the recovery behavior (Fig. 8) can be summarized in the following equation:

$$\vec{X}_i^{t+1} = \begin{cases} \text{Eq. (13)}, & if\ \tau_7 < \tau_8 \\ \text{Eq. (15)}, & otherwise \end{cases} \tag{16}$$

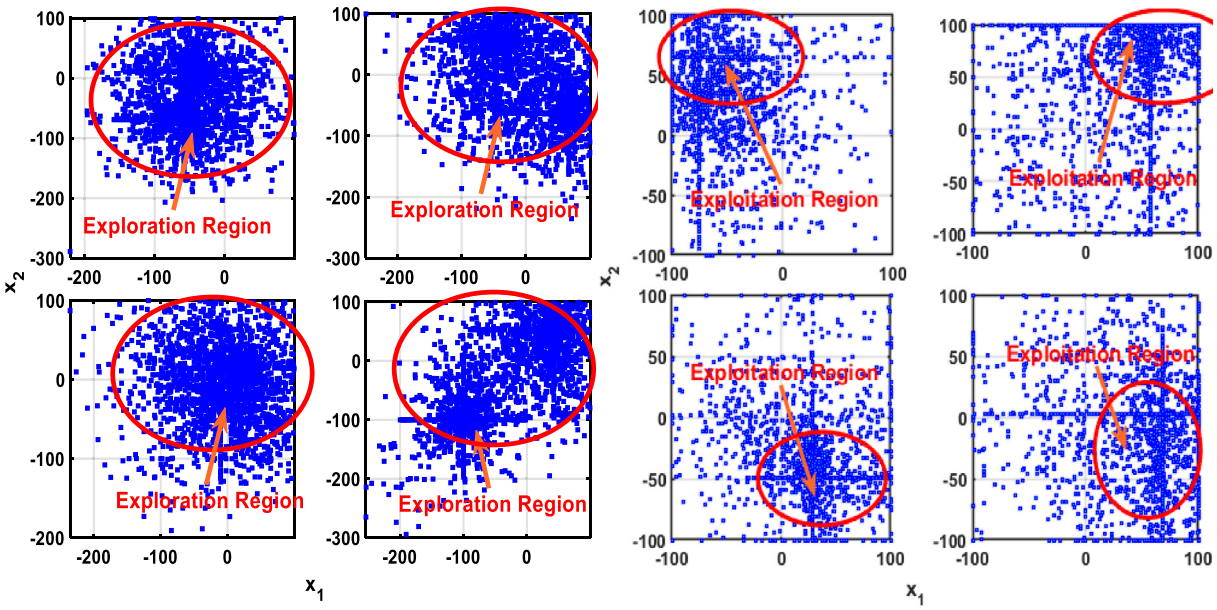
where τ_7 and τ_8 are random numbers in the interval [0,1]. The first case in the above equation represents a nutcracker that remembers the hidden store, while the second case represents a nutcracker that does not remember the hidden store. Now, the trade-off between exploration behaviors about the first and second RPs is achieved according to the following equation:

$$\vec{X}_i^{t+1} = \begin{cases} \text{Eq. (12)}, & if\ f(\text{Eq. (12)}) < f(\text{Eq. (14)}) \\ \text{Eq. (14)}, & otherwise \end{cases} \tag{17}$$

Finally, the exchange between the cache-search stage and the recovery stage is applied according to the following formula to maintain a balance between exploration and exploitation:

$$\vec{X}_i^{t+1} = \begin{cases} \text{Eq. (16)}, & if\ \phi > P_{a_2} \\ \text{Eq. (17)}, & otherwise \end{cases} \tag{18}$$

where ϕ is a random number between 0 and 1, and P_{a_2} represents a probability value that is equal to 0.2, and this value was established from the experiments conducted later. According to the simulation experiments conducted to show the characteristics of the cache-search and recovery strategy and reported in Fig. 9, on one hand, the exploration operator within this strategy shows strong coverage, distributing capability around wide



a) The exploration operator b) The exploitation operator

Fig. 9. Simulation of the exploration and exploitation operators in the cache-search and recovery strategy.

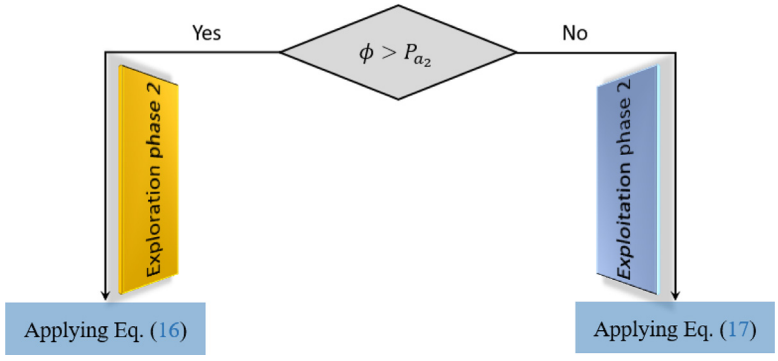


Fig. 10. Flowchart of the exploration and exploitation process in the second proposed strategy.

regions that represent the RPs following by the nutcrackers for finding the hidden caches. On the other hand, the exploitation operator pictured in Fig. 9(b) narrows the search within a small region to recover the pine seeds buried in the hidden cache. The flowchart of the exploration and exploitation processes in the second proposed strategy is depicted in Fig. 10, and listed in Algorithm 2.

Algorithm 2 *Cache-search and recovery strategy*

Output: $\vec{X}_{best}$
1. Initialize N nutcrackers using Eq.(19)
2. Evaluate each X_i and finding the one with the best fitness in $\vec{X}_{best}$
3. $t = 1$; //the current iteration
4. **while** $(t < T_{max})$
5. Generate RP matrix using Eq. (5), Eq.(9) and Eq.(10).
6. Generate a random number ϕ between 0 and 1.
7. **for** i=1:N
8. **if** $\phi > P_{a_2}$ /***Exploration phase2*/**
9. Updating $\vec{X}_i^{t+1}$ using Eq. (16) and Eq. (20)
10. **Else** /***Exploitation phase2*/**
11. Updating $\vec{X}_i^{t+1}$ using Eq. (17) and Eq. (20)
12. **End if**
13. $t = t + 1$
14. **End for**
15. **End while**

4. Implementation of the proposed method for optimization

The implementation of NOA is presented in this section. The proposed NOA is designed based on two main strategies. The first approach is the foraging and storage strategy, and the second one is the cache-search and recovery strategy. Each strategy simulates a time of nutcracker behavior. The two strategies equally important to NOA, which have the same probability to be selected. Similar to other population-based optimization algorithms, the population in NOA is initialized by

$$\vec{X}_{i,j}^{t} = + \left(\vec{U}_j - \vec{L}_j\right) \cdot \overrightarrow{RM} + \vec{L}_j,\ I = 1, 2, \ldots, N,\ j = 1, 2, \ldots, D \quad (19)$$

where t denotes the generation index; i indicates the population index; $\vec{U}_j$ and $\vec{L}_j$ are the upper and lower boundaries for a D-dimensional problem, respectively; and $\overrightarrow{RM}$ is a random vector in the interval [0, 1]. Each nutcracker represents a feasible solution to the problem.

The first proposed strategy is based on the designed exploration and exploitation stages called the first exploration and exploitation phases, respectively. The second proposed strategy is based on the designed exploration and exploitation stages called the second exploration and exploitation phases, respectively. In both strategies, the exploration and exploitation phases have the same importance to NOA; thus, they are randomly exchanged to

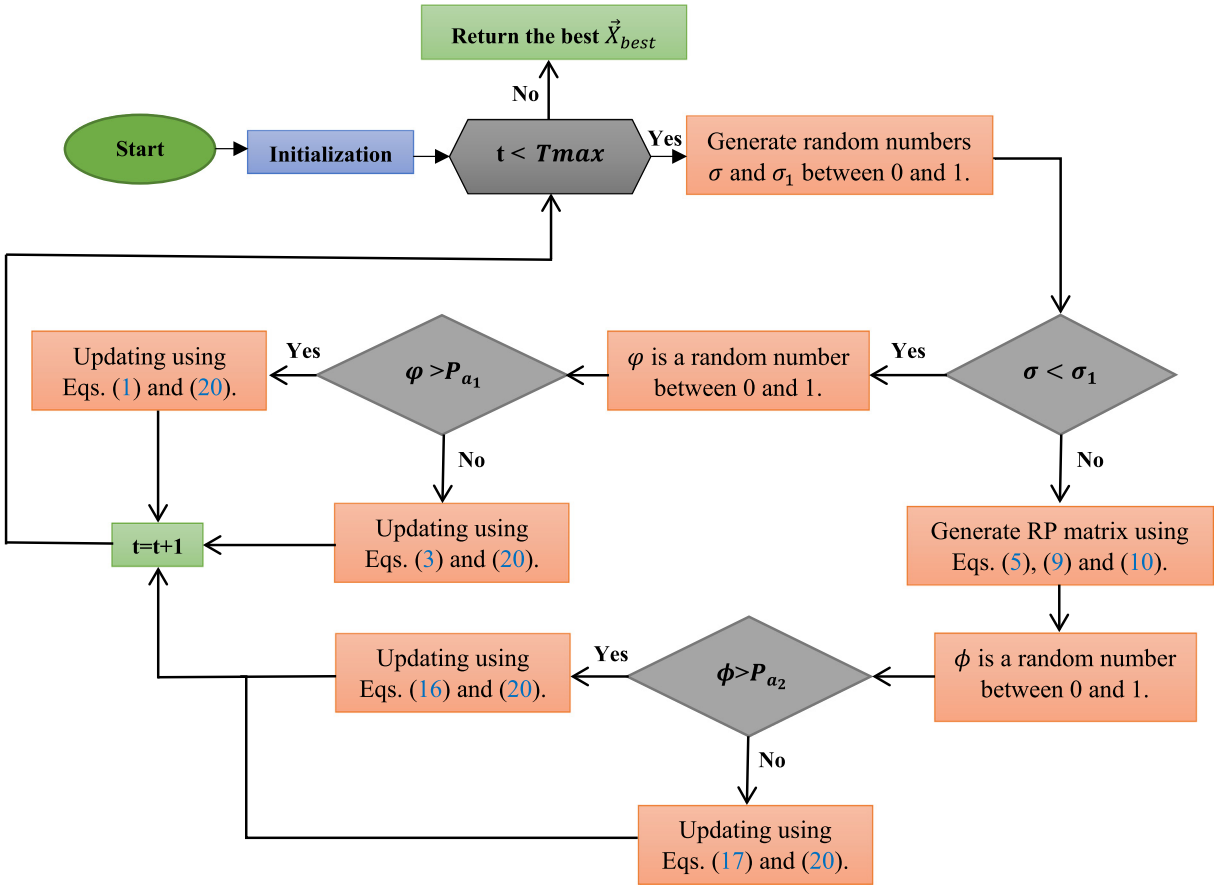


Fig. 11. Flowchart of NOA.

simulate the behavior of the nutcrackers while applying the first or second proposed strategy according to their need.

In the first proposed strategy, each nutcracker/source represents a potential solution in the current iteration, and every cache represents a potential solution in that iteration. In the second proposed strategy, each nutcracker/cache represents a potential solution in the current iteration, and every RP represents a potential solution in that iteration. Each nutcracker has one food source and one cache in search space. The quality of the solution is evaluated for each new position of the nutcracker based on a fitness function. Then, the current position is updated if the solution quality of the new position is better than the solution quality of the current one. However, the nutcrackers in the NOA model remain at their current position if their solution quality is higher than the new one. In summary, the above concept can be expressed by Eq. (17). Finally, the pseudo-code and flowchart of NOA are shown in Algorithm 3 and Fig. 11, respectively. The time complexity of this algorithm in Big-O according to the steps listed is $O(T_{max}N)$. The advantages of NOA are listed as follows:

- Easy to implement
- Able to avoid falling into local optima for several optimization problems with various characteristics
- Having a high convergence speed.

While its main limitation considered as one of our future work is listed as follows:

- Inability to balance its exploration and exploitation according to the needs of the optimization problem to save computational cost and avoid local optimization entrapment.

$$\vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^{t+1}, & if\ f(\vec{X}_i^{t+1}) < f(\vec{X}_i^{t}) \\ \vec{X}_i^{t}, & otherwise \end{cases} \tag{20}$$

5. Results and discussion

This section first investigates the performance of the proposed optimizer to show its efficiency for 23 well-known mathematical optimization problems employed widely in the literature to measure the exploitation and exploration capabilities of the newly-proposed optimizers [55,87]. These test functions belong to three categories: unimodal which contains only one global optimum to investigate the exploitation operator of an optimization algorithm; high-dimensional multimodal which involves several local minima to observe the exploration operator; and fixed-dimensional multimodal which involves a limited number of dimensions to simulate the nature of the real-world optimization problems like the parameter estimation of PV models and fuel cell, and several else [88–90]. The characteristics, including the number of dimensions (D), search domain, and global optima, along with the mathematical model of these test functions, are presented in Appendix. Then, Three recent and challenging test suites on numerical optimization competitions: CEC-2014, CEC-2017, and CEC-2020 are utilized in this study to further validate the performance of NOA [91–93]. These benchmarks contain 30 test functions that are unimodal, multimodal, hybrid, and composition in nature, and they are designed to strongly challenge the previously proposed and the newly proposed metaheuristic algorithms. The characteristics of these test suites are described in Appendix.

Algorithm 3 Pseudo-code of NOA

Input: population size N, the lower limits of variables $\vec{L}$, the upper limits of variables $\vec{U}$ the current number of iteration t=0, and the maximum number of iterations T_{max};
Output: the best solution found

```
Initialize N nutcracker/solution using Eq.(19);
Evaluate each solution and find the one with the best fitness in the population
t = 1; //the current function evaluation//
while (t < T_max)
    Generate random numbers σ and σ_1 between 0 and 1.
    If σ < σ_1  //* Foraging and storage strategy*//
        φ is a random number between 0 and 1.
        for i=1:N
            for j= 1:d
                if φ >P_a1 /*Exploration phase1*/
                    Updating X_i^{t+1} using Eq. (1) and Eq. (20)
                else  /*Exploitation phase1*/
                    Updating X_i^{t+1} using Eq. (3) and Eq. (20)
                end if
            end for
            Update the current iteration t by t = t + 1
        end for
    else //* Cache-search and recovery strategy *//
        Generate RP matrix using Eq. (5), Eq.(9) and Eq.(10).
        Generate a random number ϕ between 0 and 1.
        for i=1:N
            if ϕ >P_a2 /*Exploration phase2*/
                Updating X_i^{t+1} using Eq. (16) and Eq. (20)
            else  /*Exploitation phase2*/
                Updating X_i^{t+1} using Eq. (17) and Eq. (20)
            end if
            t = t + 1
        end for
end while
```

To reveal the efficiency of the proposed NOA, it is extensively compared with three classes of existing optimization algorithms: (1) the first class includes some of the recently-published algorithms like gradient-based optimizer (GBO, 2020) [94], artificial gorilla troops optimizer (GTO, 2021) [27], Runge Kutta method (RUN, 2021) beyond the metaphor [95], African vultures optimization algorithm (AVOA, 2021) [96], equilibrium optimizer (EO, 2020) [55], red Fox optimizer (RFO, 2021) [26], and slime mould algorithm (SMA, 2021) [87], (2) the second class contains some of the highly-cited, well-studied algorithms such as whale optimization algorithm (WOA, 2016) [17], gray wolf optimizer (GWO, 2014) [14], and salp swarm algorithm (SSA, 2017) [18] and (3) the last class includes AL-SHADE [97], L-SHADE [98], LSHADE-cnEpSin [92], and LSHADE-SPACMA [99] as highly-performing optimizers and winners of CEC competition. All of these competing algorithms were put forth to address the global optimization problem solved in this study, so we used the controlling parameter values in our experiments as recommended by the authors in the cited paper, with the exception of N and T_{max}, which were set to 25 and 50000 to ensure an equal comparison. Table 2 presents the controlling parameter values used in the conducted experiments for each rival algorithm.

All algorithms used in our experiments were run independently 30 times and their optimization processes continued within each run until a function evaluation of 50,000 was reached. The results of those runs are analyzed to determine the average and standard deviation (Std) of the fitness values obtained within these runs, as well as the rank metric to determine the order of each algorithm in terms of the average fitness value. Additionally, the Wilcoxon rank-sum test and convergence curve metrics are used to demonstrate the significant difference between NOA's results and those of rival algorithms, as well as the fastest algorithm to the global solution. The experiments were carried out using MATLAB R2019a on a device with the following specifications: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40 GHz, 32 GB RAM, and 64-bit Windows 10 Pro.

The proposed NOA has three main effective controlling parameters which have to be accurately optimized to reach the best performance; these parameters are P_{a_1} to determine the percentage of updating the current solution in the regions out of the current population for covering intractable regions, P_{a_2} to determine the probability of exchanging between the cache-search stage and the recovery stage, and δ to determine if the current dimension will be updated within the upper and lower bound of the optimization problem as an attempt to avoid stuck into local minima. To pick the optimal values for these parameters, extensive experiments under various values for each parameter have been done and their conclusion is introduced in Figs. 12–14, which show that the best values for these parameters for the test functions F60 and F61 are 0.2, 0.4, and 0.05, respectively.

5.1. Exploration and exploitation analysis

First, the unimodal test functions (F1–F6) which have only a global best position are herein employed to evaluate the proposed algorithm's exploitative behavior to demonstrate its ability to attack these global best solutions. Table 3 compares NOA's performance to that of 10 other rival algorithms, including SSA, GBO, RUN, WOA, GTO, AVOA, EO, GWO, RFO, and SMA for six unimodal test functions in terms of the average (Ave), and standard deviation (Std) of 30 fitness values. NOA is competitive to most strong optimizers like RFO, SMA, GTO, AVOA, GBO, and RUN for the first four unimodal test functions, and unfortunately inferior to four and six optimizers for F5 and F6, respectively, as shown in the ranking row.

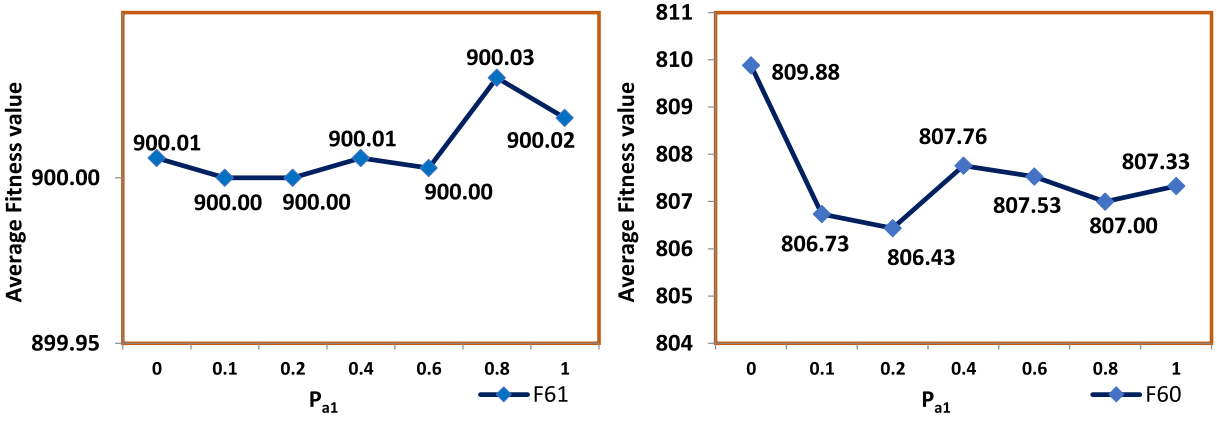
Also, this section evaluates NOA's exploratory behavior on multimodal test functions (F7–F23), which contain multiple local minima. The average (Ave), and standard values of the fitness values obtained by the proposed and compared optimizers over 30 independent runs are shown in Table 3. The results in this table demonstrate that NOA is competitive with some of the compared optimizers in terms of Ave for the majority of the multimodal test functions, inferior for three test functions (F7, F8, and F13), and superior to all rival methods in terms of Std values for most multimodal test functions. The comparison in this section demonstrates that NOA has an excellent exploratory operator due to its diverse exploratory behaviors at various stages of the optimization process, in addition to having a strong exploitation operator to explore these solutions found around the best-so-far solution.

5.2. Comparison over CEC2014

On the CEC-2014 test suite, additional validation is carried out to ensure that the introduced and other methods perform as expected. This suite of test functions is used to evaluate an algorithm's ability to explore, escape from local minima, and exploit. It is divided into unimodal, multimodal, hybrid, and composition categories. Appendix contains a description of the characteristics of this test suite, in addition to the considered dimensions which are 10. Table 4 presents the Ave, and Std values in addition to the rank metric, which demonstrates that NOA is the best because it ranks first among all rival algorithms on all CEC-2014 test functions except for F27, F30, F51, and F53. On all CEC-2014 test functions, the Ave rank row as the second row from the last within Table 4 affirms that NOA comes in the 1st rank with a value of 1.52, followed by GBO with a value of 4.48, while RFO comes in the last rank. The last row in this table shows the average standard deviation (Std) of all CEC-2014 test functions; this row demonstrates that the outcomes obtained by NOA are significantly similar since it has the lowest Std value of 13.7, followed by SMA as the second-best one. The results of the

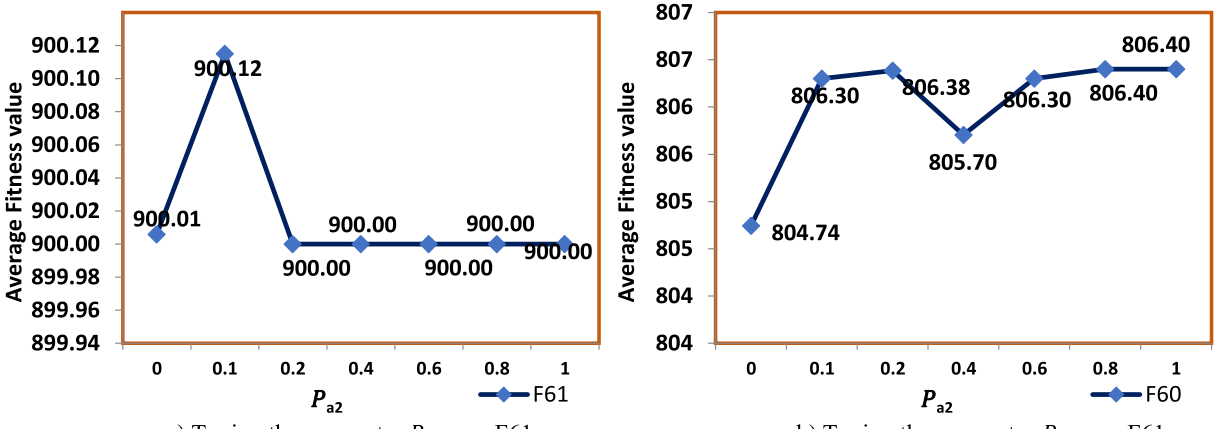
Table 2
Parameter settings of the competitors and proposed NOA.

Algorithms	Parameters	Value	Algorithms	Parameters	Value
GWO (2014)	Convergence constant a N	Decreases Linearly from 2 to 0 25	SMA (2020)	z N	0.03 25
WOA (2017)	Convergence constant a Spiral factor b N	Decreases Linearly from 2 to 0 1 25	GTO (2021)	p $Beta$ w N	0.03 3 8 25
EO (2020)	a_1 a_2 V GP N	2 1 1 0.5 25	AVOA (2021)	Alpha (L_1) Beta (L_2) Gamma (w) P_1 P_2 P_3 N	0.8 0.2 2.5 0.6 0.4 0.6 25
RUN (2021)	a b N	20 12 25	RFO (2021)	N	25
GBO(2020)	pr β_{min} β_{max} N	0.5 0.2 1.2 25	NOA (Proposed)	P_{rp} P_{a_2} δ N	0.2 0.4 0.05 25
SSA (2017)	c1 N	Decreases from 2 to 0 25			



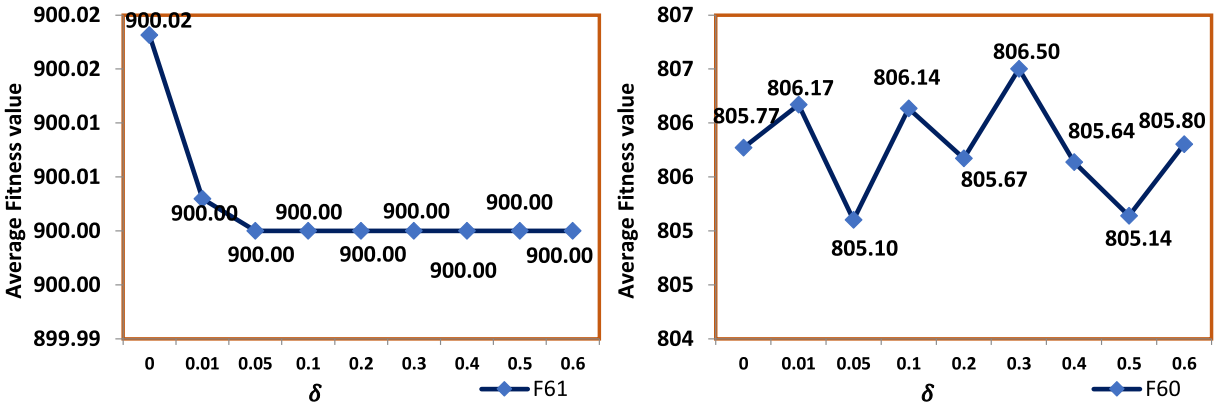
a) Tuning the parameter P_{a_1} over F61 b) Tuning the parameter P_{a_1} over F61

Fig. 12. Sensitivity analysis of the parameter P_{a_1}.



a) Tuning the parameter P_{a_2} over F61 b) Tuning the parameter P_{a_2} over F61

Fig. 13. Sensitivity analysis of the parameter P_{a_2}.



a) Tuning the parameter δ over F61 b) Tuning the parameter δ over F61

Fig. 14. Sensitivity analysis of the parameter δ.

Wilcoxon rank-sum statistical test for most test functions are less than 0.05, which confirms that the outcomes obtained by NOA are significantly different when compared to those obtained by the competitors, as shown in Table 5.

5.3. Comparison over CEC2017

The CEC-2017 test suite is used in this section to compare the performance of NOA and other optimizers [92]. It includes four families of mathematical functions: unimodal (F54–F55), multimodal (F56–F62), composition (F63–F72), and hybrid (F73–F82). As previously stated, unimodal test functions are preferable for assessing the exploitability of optimization methods because they contain only one global optimal solution; multimodal test functions contain multiple locally optimal solutions, making them particularly well-suited for assessing the exploratory capabilities of newly introduced optimizers, and both composition and hybrid functions have been designed to assess the methods' ability to avoid local optima. For all benchmark functions discussed in this section, the dimension is set to 10. Appendix shows the CEC-2017 benchmark's specifications. Table 6 shows the Ave, Std, *p*-value, and Rank values achieved by the proposed and rival optimizers in 30 independent runs on this suite. This table shows that NOA is the best optimizer for all unimodal, multimodal, and hybrid test functions, and superior for 8 out of 10 multimodal test functions. Specifically, NOA could be superior for 27 out of 29 CEC-2017 test functions.

The last two rows in Table 6 display the average of the rank values and Std values on all test functions for each algorithm, respectively. According to these rows, NOA has the potential to be the best with a value of 1.14 and 24.4 for average rank and Std, respectively. The Wilcoxon rank-sum test is utilized to calculate the *p*-value between NOA and each rival algorithm on each CEC-2017 test function to determine the difference. Wilcoxon rank-sum test reveal that NOA has significantly-different outcomes from these of the rival optimizers since the p-values shown in Table 7 accept the alternative hypothesis which supposes that there are the difference between the outcomes of the proposed and these of a rival algorithm. The experiments conducted in this section show that NOA belongs to the category of the strong optimizers because it could defeat GBO, RUN, GTO, AVOA, SMA, RFO, and EO as the most recent published optimizers as well as four well-established metaheuristic optimizers like WOA, GWO, and SSA.

5.4. Comparison over CEC2020

Further testing on the CEC-2020 test suite is carried out here to verify the performance of the suggested and other methods. An algorithm's ability to explore, exploit and avoid local minima can be tested utilizing this suite which contains 10 test functions and is divided into unimodal, multimodal, hybrid, and compositional. In Appendix, the properties of this test suite are laid out, and their dimensions are 10. Table 8 presents the ave, rank, and Std values obtained by analyzing the outcomes of 30 independent runs, which shows that NOA has superior performance for all test functions found in the CEC-2020 test suite. The average of the rank values and the Std values on all test functions for each algorithm are shown in the last two rows of Table 8, respectively. According to these rows, NOA has the potential to be the best, with average rank and standard deviation values of 1.00 and 28.2, respectively. Additionally, It is necessary to calculate the *p*-value between NOA and each rival algorithm on each CEC-2020 test function to determine the difference between the two algorithms. The results of the Wilcoxon rank-sum tests reveal that NOA produces significantly different results from those produced by rival optimizers, as shown in Table 9. The p-values in Table 9 support the alternative hypothesis, which assumes that there is a difference between the outcomes of the proposed algorithm and those produced by a rival algorithm. This section presents additional experimental evidence that NOA belongs to the category of strong optimizers.

5.5. Convergence speed analysis

In this section, the convergence speed of NOA is compared to that of the rival algorithms to demonstrate their differences (see Fig. 15). This figure shows that all NOA convergence curves have an accelerated reducing pattern, which is particularly noticeable in the last half of the optimization process for four families of test functions (unimodal, multimodal, composition, and hybrid). According to the convergence curves shown in Fig. 15, the NOA optimizer is significantly faster than all of the other competing algorithms. This is due to its capabilities to balance between exploration and exploitation operators which aid in voiding stagnation into local minima with accelerating convergence speed in the right direction of the most promising areas obtained by far, especially in the last half of the optimization process.

5.6. Qualitative analysis

Diversity, convergence curve, average fitness value, trajectory in 1st dimension, and search history are all metrics used in this section to describe the NOA performance during the optimization process. An individual's average distance from other individuals is shown by the diversity metric; the convergence curve shows the best-fitness value achieved within each iteration; the average

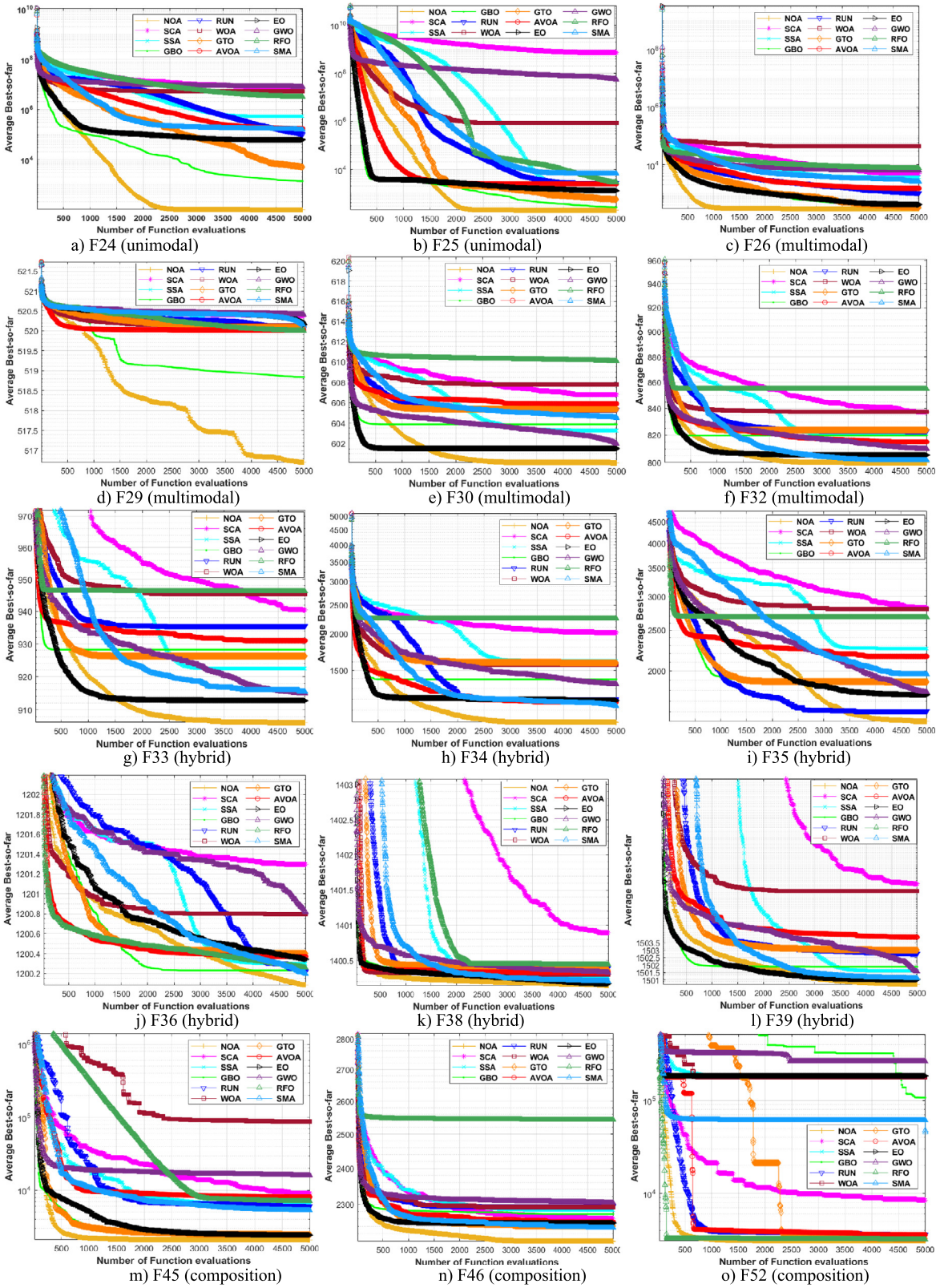


Fig. 15. Averaged Convergence speed for rival optimizers and proposed on some test functions.

Table 3
Results of unimodal and multimodal test functions.

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Unimodal												
F1	Ave	**0.00 × 10^0**	1.03 × 10^−6	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	2.58 × 10^−124	7.51 × 10^−60	**0.00 × 10^0**	**0.00 × 10^0**
	Std	**0.00 × 10^0**	5.29 × 10^−8	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	3.65 × 10^−124	7.56 × 10^−61	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	4	**1**	**1**	**1**	**1**	**1**	2	3	**1**	**1**
F2	Ave	**0.00 × 10^0**	1.02 × 10^1	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	2.46 × 10^−73	9.20 × 10^−36	**0.00 × 10^0**	**0.00 × 10^0**
	Std	**0.00 × 10^0**	4.07 × 10^0	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	2.70 × 10^−73	4.14 × 10^−36	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	4	**1**	**1**	**1**	**1**	**1**	2	3	**1**	**1**
F3	Ave	**0.00 × 10^0**	4.08 × 10^4	**0.00 × 10^0**	**0.00 × 10^0**	4.85 × 10^5	**0.00 × 10^0**	**0.00 × 10^0**	9.87 × 10^−13	8.70 × 10^−4	**0.00 × 10^0**	**0.00 × 10^0**
	Std	**0.00 × 10^0**	1.48 × 10^3	**0.00 × 10^0**	**0.00 × 10^0**	3.02 × 10^4	**0.00 × 10^0**	**0.00 × 10^0**	1.04 × 10^−12	1.21 × 10^−3	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	4	**1**	**1**	5	**1**	**1**	2	3	**1**	**1**
F4	Ave	**0.00 × 10^0**	2.61 × 10^1	**0.00 × 10^0**	**0.00 × 10^0**	5.19 × 10^1	**0.00 × 10^0**	**0.00 × 10^0**	1.41 × 10^−22	1.55 × 10^−7	**0.00 × 10^0**	**0.00 × 10^0**
	Std	**0.00 × 10^0**	1.40 × 10^0	**0.00 × 10^0**	**0.00 × 10^0**	4.56 × 10^1	**0.00 × 10^0**	**0.00 × 10^0**	1.01 × 10^−22	2.02 × 10^−7	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	4	**1**	**1**	5	**1**	**1**	2	3	**1**	**1**
F5	Ave	9.46 × 10^1	4.12 × 10^2	8.78 × 10^1	9.76 × 10^1	9.74 × 10^1	3.46 × 10^−4	**3.68 × 10^−6**	9.47 × 10^1	9.72 × 10^1	9.87 × 10^1	1.17 × 10^0
	Std	6.29 × 10^−1	2.40 × 10^2	1.10 × 10^0	6.19 × 10^−1	6.48 × 10^−2	1.66 × 10^−4	**1.63 × 10^−6**	9.50 × 10^−1	1.61 × 10^0	3.61 × 10^−1	1.36 × 10^0
	Rank	5	11	4	9	8	2	**1**	6	7	10	3
F6	Ave	2.69 × 10^−2	**5.69 × 10^−7**	5.06 × 10^−5	7.39 × 10^−7	8.51 × 10^−1	2.52 × 10^−4	1.03 × 10^−6	5.00 × 10^−1	9.21 × 10^0	4.33 × 10^−1	1.23 × 10^−2
	Std	1.06 × 10^−2	**6.44 × 10^−8**	1.93 × 10^−5	7.93 × 10^−7	4.81 × 10^−1	1.70 × 10^−4	3.42 × 10^−7	6.70 × 10^−3	3.16 × 10^−1	1.85 × 10^−2	1.81 × 10^−3
	Rank	7	**1**	4	2	10	5	3	9	**11**	8	**6**
High-dimensional multimodal												
F7	Ave	6.20 × 10^−5	9.52 × 10^−1	1.00 × 10^−4	8.18 × 10^−5	8.29 × 10^−4	**2.58 × 10^−5**	4.81 × 10^−5	7.45 × 10^−4	1.13 × 10^−3	5.02 × 10^−5	7.81 × 10^−5
	Std	6.77 × 10^−5	7.32 × 10^−2	2.49 × 10^−5	4.66 × 10^−5	1.02 × 10^−3	**4.16 × 10^−6**	4.94 × 10^−5	4.26 × 10^−4	3.18 × 10^−4	6.75 × 10^−5	1.97 × 10^−5
	Rank	**4**	11	7	5	9	**1**	2	8	10	3	6
F8	Ave	−3.69 × 10^4	−2.39 × 10^4	−2.68 × 10^4	−2.26 × 10^4	−3.88 × 10^4	−4.19 × 10^4	−4.18 × 10^4	−3.03 × 10^4	−1.60 × 10^4	−2.11 × 10^4	−4.19 × 10^4
	Std	4.38 × 10^3	1.31 × 10^3	1.16 × 10^3	1.80 × 10^3	4.59 × 10^2	**1.14 × 10^−2**	8.37 × 10^1	5.94 × 10^2	7.02 × 10^2	1.61 × 10^2	5.80 × 10^−2
	Rank	5	8	7	9	4	**1**	3	6	11	10	2
F9	Ave	**0.00 × 10^0**	1.84 × 10^2	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**
	Std	**0.00 × 10^0**	8.44 × 10^0	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F10	Ave	**8.88 × 10^−16**	5.52 × 10^0	**8.88 × 10^−16**	**8.88 × 10^−16**	**8.88 × 10^−16**	**8.88 × 10^−16**	**8.88 × 10^−16**	6.22 × 10^−15	3.11 × 10^−14	**8.88 × 10^−16**	**8.88 × 10^−16**
	Std	**0.00 × 10^0**	2.55 × 10^0	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	2.51 × 10^−15	2.51 × 10^−15	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	4	**1**	**1**	**1**	**1**	**1**	2	3	**1**	**1**
F11	Ave	**0.00 × 10^0**	2.36 × 10^−2	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**
	Std	**0.00 × 10^0**	5.81 × 10^−4	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**
	Rank	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F12	Ave	1.18 × 10^−4	1.07 × 10^1	1.70 × 10^−6	2.12 × 10^−8	1.01 × 10^−2	1.25 × 10^−6	**3.91 × 10^−9**	1.56 × 10^−2	2.05 × 10^−1	1.03 × 10^−2	6.45 × 10^−5
	Std	1.48 × 10^−5	4.90 × 10^0	1.05 × 10^−6	6.64 × 10^−10	3.91 × 10^−3	1.15 × 10^−6	**3.05 × 10^−9**	2.20 × 10^−2	2.79 × 10^−2	2.61 × 10^−3	8.57 × 10^−5
	Rank	**6**	11	4	2	7	3	**1**	9	10	8	5
F13	Ave	2.80 × 10^−2	1.63 × 10^2	3.09 × 10^0	2.32 × 10^−2	1.37 × 10^0	1.70 × 10^−5	**1.68 × 10^−9**	3.38 × 10^0	5.98 × 10^0	5.25 × 10^0	7.12 × 10^−3
	Std	4.59 × 10^−3	1.11 × 10^0	4.37 × 10^0	1.37 × 10^−2	8.32 × 10^−1	2.24 × 10^−5	**5.02 × 10^−10**	2.62 × 10^0	4.32 × 10^−1	6.61 × 10^0	3.29 × 10^−3
	Rank	**5**	11	7	4	6	2	**1**	10	9	8	3
Fixed-dimensional multimodal												
F14	Ave	**9.98 × 10^−1**	9.98 × 10^−1	**9.98 × 10^−1**	6.87 × 10^0	9.98 × 10^−1	**9.98 × 10^−1**	**9.98 × 10^−1**	**9.98 × 10^−1**	6.87 × 10^0	1.27 × 10^1	9.98 × 10^−1
	Std	**0.00 × 10^0**	1.57 × 10^−16	**0.00 × 10^0**	5.50 × 10^0	8.99 × 10^−14	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	5.50 × 10^0	2.12 × 10^−13	1.08 × 10^−14
	Rank	**1**	2	**1**	6	4	**1**	**1**	**1**	6	7	3
F15	Ave	**3.07 × 10^−4**	1.00 × 10^−3	**3.07 × 10^−4**	7.65 × 10^−4	5.22 × 10^−4	**3.07 × 10^−4**	3.08 × 10^−4	3.08 × 10^−4	1.03 × 10^−2	9.51 × 10^−4	3.14 × 10^−4
	Std	**7.67 × 10^−20**	3.15 × 10^−4	7.67 × 10^−20	6.47 × 10^−4	3.00 × 10^−4	4.23 × 10^−19	2.87 × 10^−7	1.43 × 10^−8	1.42 × 10^−2	9.10 × 10^−4	8.42 × 10^−6
	Rank	**1**	8	**1**	6	5	**1**	2	2	9	7	3
F16	Ave	**−1.03 × 10^0**	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0	−1.03 × 10^0
	Std	**0.00 × 10^0**	4.97 × 10^−16	0.00 × 10^0	1.32 × 10^−14	8.02 × 10^−12	0.00 × 10^0	2.22 × 10^−16	2.22 × 10^−16	9.14 × 10^−10	6.46 × 10^−11	2.05 × 10^−15
	Rank	**1**	1	1	1	1	1	1	1	1	1	1
F17	Ave	**3.98 × 10^−1**	3.98 × 10^−1	3.98 × 10^−1	3.98 × 10^−1	3.98 × 10^−1	**3.98 × 10^−1**	**3.98 × 10^−1**	**3.98 × 10^−1**	3.98 × 10^−1	3.98 × 10^−1	3.98 × 10^−1
	Std	**0.00 × 10^0**	2.14 × 10^−14	0.00 × 10^0	3.41 × 10^−11	1.35 × 10^−7	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	1.30 × 10^−8	2.83 × 10^−11	6.50 × 10^−9
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F18	Ave	**3.00 × 10^0**	3.00 × 10^0	3.00 × 10^0	3.00 × 10^0	3.00 × 10^0	**3.00 × 10^0**	3.00 × 10^0	3.00 × 10^0	3.00 × 10^0	3.00 × 10^0	3.00 × 10^0
	Std	**0.00 × 10^0**	4.58 × 10^−14	4.44 × 10^−16	6.56 × 10^−14	3.26 × 10^−8	**0.00 × 10^0**	5.71 × 10^−8	9.93 × 10^−16	1.35 × 10^−6	2.66 × 10^−9	8.23 × 10^−14
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F19	Ave	**−3.86 × 10^0**	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0	−3.86 × 10^0
	Std	**0.00 × 10^0**	9.74 × 10^−15	0.00 × 10^0	6.80 × 10^−12	3.11 × 10^−3	0.00 × 10^0	0.00 × 10^0	0.00 × 10^0	3.89 × 10^−3	1.99 × 10^−8	2.65 × 10^−9
	Rank	**1**	2	1	1	1	1	1	1	1	1	1
F20	Ave	**−3.32 × 10^0**	−3.26 × 10^0	−3.26 × 10^0	−3.32 × 10^0	−3.16 × 10^0	−3.32 × 10^0	−3.26 × 10^0	−3.26 × 10^0	−3.26 × 10^0	−3.26 × 10^0	−3.20 × 10^0
	Std	**0.00 × 10^0**	8.43 × 10^−2	8.41 × 10^−2	4.20 × 10^−10	5.60 × 10^−2	0.00 × 10^0	8.41 × 10^−2	8.41 × 10^−2	8.52 × 10^−2	8.41 × 10^−2	7.37 × 10^−8
	Rank	**1**	2	2	**1**	4	**1**	2	2	2	2	3
F21	Ave	**−1.02 × 10^1**	−7.63 × 10^0	**−1.02 × 10^1**	**−1.02 × 10^1**	**−1.02 × 10^1**	**−1.02 × 10^1**	**−1.02 × 10^1**	**−1.02 × 10^1**	−7.63 × 10^0	−5.08 × 10^0	**−1.02 × 10^1**
	Std	**0.00 × 10^0**	3.57 × 10^0	**0.00 × 10^0**	**6.42 × 10^−12**	**3.78 × 10^−4**	**0.00 × 10^0**	**0.00 × 10^0**	**0.00 × 10^0**	3.57 × 10^0	3.22 × 10^−2	**5.13 × 10^−5**
	Rank	**1**	4	**1**	**1**	**1**	**1**	**1**	**1**	4	3	**1**
F22	Ave	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	**−1.04 × 10^1**	−5.09 × 10^0	**−1.04 × 10^1**
	Std	**1.78 × 10^−15**	**1.15 × 10^−12**	**0.00 × 10^0**	**1.60 × 10^−10**	**5.69 × 10^−4**	**1.78 × 10^−15**	**0.00 × 10^0**	**0.00 × 10^0**	**2.79 × 10^−5**	2.34 × 10^−7	**1.65 × 10^−5**
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	3	**1**
F23	Ave	**−1.05 × 10^1**	−6.67 × 10^0	**−1.05 × 10^1**	**−1.05 × 10^1**	**−1.05 × 10^1**	**−1.05 × 10^1**	**−1.05 × 10^1**	**−1.05 × 10^1**	**−1.05 × 10^1**	−5.13 × 10^0	**−1.05 × 10^1**
	Std	**0.00 × 10^0**	5.47 × 10^0	**0.00 × 10^0**	**5.45 × 10^−11**	**1.36 × 10^−4**	**0.00 × 10^0**	**0.00 × 10^0**	**3.97 × 10^−15**	**1.73 × 10^−5**	2.36 × 10^−7	**7.69 × 10^−6**
	Rank	**1**	4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	3	**1**

Bold values indicate the best findings.

fitness value shows the average of all individuals' fitness values throughout every one of those iterations; the trajectory curve shows how the 1st dimension of a solution is changed over time as it moves through the optimization process; the search history discloses the positions attained by the NOA optimizer during the optimization process.

Fig. 16 shows that NOA's diversity metric decreases over time, as shown in the second column, indicating that NOA's efficiency

Table 4
Results of unimodal, multimodal, hybrid, and composition CEC-2014 test functions.

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Unimodal												
F24	Ave	**1.00 × 10²**	4.18 × 10⁵	8.51 × 10³	1.24 × 10⁵	4.95 × 10⁶	5.09 × 10³	1.38 × 10⁵	5.20 × 10⁴	7.27 × 10⁶	2.88 × 10⁶	1.38 × 10⁵
	Std	**8.11 × 10⁻⁶**	3.66 × 10⁵	3.78 × 10⁴	1.23 × 10⁵	4.65 × 10⁶	5.68 × 10³	1.18 × 10⁵	6.65 × 10⁴	4.92 × 10⁶	4.25 × 10⁶	5.88 × 10⁴
	Rank	**1**	8	3	5	10	2	6	4	11	9	7
F25	Ave	**2.00 × 10²**	3.10 × 10³	2.57 × 10²	3.37 × 10³	8.41 × 10⁵	9.91 × 10²	3.34 × 10³	1.43 × 10³	1.73 × 10⁷	3.20 × 10³	7.34 × 10³
	Std	**7.29 × 10⁻¹¹**	2.91 × 10³	9.23 × 10¹	5.07 × 10³	6.03 × 10⁵	1.76 × 10³	3.77 × 10³	1.44 × 10³	6.59 × 10⁷	2.99 × 10³	3.47 × 10³
	Rank	**1**	5	2	8	10	3	7	4	11	6	9
F26	Ave	**3.00 × 10²**	2.87 × 10³	3.12 × 10²	9.67 × 10²	4.92 × 10⁴	3.08 × 10²	1.16 × 10³	4.39 × 10²	5.87 × 10³	1.30 × 10⁴	2.39 × 10³
	Std	**2.19 × 10⁻¹³**	2.06 × 10³	2.14 × 10¹	3.80 × 10²	2.91 × 10⁴	1.14 × 10¹	5.52 × 10²	2.03 × 10²	3.95 × 10³	1.51 × 10⁴	2.53 × 10³
	Rank	**1**	8	3	5	12	2	6	4	10	11	7
Multimodal												
F27	Ave	4.25 × 10²	4.25 × 10²	4.20 × 10²	**4.18 × 10²**	4.41 × 10²	4.23 × 10²	4.19 × 10²	4.27 × 10²	4.30 × 10²	4.32 × 10²	4.21 × 10²
	Std	1.54 × 10¹	1.55 × 10¹	1.66 × 10¹	**1.60 × 10¹**	2.78 × 10¹	1.63 × 10¹	1.90 × 10¹	1.41 × 10¹	2.04 × 10¹	1.69 × 10¹	1.63 × 10¹
	Rank	7.00	6.00	3.00	**1.00**	11.00	5.00	2.00	8.00	9.00	10.00	4.00
F28	Ave	**5.18 × 10²**	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.20 × 10²	5.19 × 10²
	Std	**6.11 × 10⁰**	7.70 × 10⁻²	9.68 × 10⁻²	1.62 × 10⁻¹	1.14 × 10⁻¹	9.22 × 10⁻²	4.72 × 10⁻²	6.74 × 10⁻²	1.04 × 10⁻¹	5.45 × 10⁻⁵	3.66 × 10⁰
	Rank	**1**	4	6	9	10	8	5	7	12	3	2
F29	Ave	**6.00 × 10²**	6.03 × 10²	6.04 × 10²	6.06 × 10²	6.08 × 10²	6.05 × 10²	6.06 × 10²	6.02 × 10²	6.02 × 10²	6.10 × 10²	6.04 × 10²
	Std	**4.07 × 10⁻¹**	1.78 × 10⁰	1.40 × 10⁰	1.37 × 10⁰	1.54 × 10⁰	1.62 × 10⁰	1.36 × 10⁰	1.49 × 10⁰	1.52 × 10⁰	1.07 × 10⁰	1.94 × 10⁰
	Rank	**1**	4	6	9	11	7	8	2	3	12	5
F30	Ave	**700.0472**	700.3388	700.1710	700.3679	701.0390	700.3071	700.3811	700.0504	702.0864	705.1798	700.2654
	Std	**3.14 × 10⁻²**	1.29 × 10⁻¹	8.69 × 10⁻²	2.29 × 10⁻¹	4.52 × 10⁻¹	3.21 × 10⁻¹	1.90 × 10⁻¹	3.32 × 10⁻²	2.78 × 10⁰	4.67 × 10⁰	1.28 × 10⁻¹
	Rank	**1**	4	3	8	9	7	5	2	10	11	6
F31	Ave	**8.00 × 10²**	8.20 × 10²	8.21 × 10²	8.23 × 10²	8.38 × 10²	8.21 × 10²	8.12 × 10²	8.05 × 10²	8.08 × 10²	8.48 × 10²	8.02 × 10²
	Std	**2.11 × 10⁻¹⁴**	6.79 × 10⁰	1.06 × 10¹	7.93 × 10⁰	1.43 × 10¹	8.75 × 10⁰	5.70 × 10⁰	2.89 × 10⁰	3.23 × 10⁰	1.84 × 10¹	1.09 × 10⁰
	Rank	**1**	6	7	9	10	8	5	3	4	12	2
F32	Ave	**9.06 × 10²**	9.20 × 10²	9.26 × 10²	9.37 × 10²	9.46 × 10²	9.32 × 10²	9.35 × 10²	9.11 × 10²	9.15 × 10²	9.54 × 10²	9.15 × 10²
	Std	**2.44 × 10⁰**	8.81 × 10⁰	1.06 × 10¹	6.95 × 10⁰	1.48 × 10¹	1.31 × 10¹	1.02 × 10¹	5.11 × 10⁰	7.15 × 10⁰	9.32 × 10⁰	7.99 × 10⁰
	Rank	**1**	5	6	9	11	7	8	2	4	12	3
F33	Ave	**1.00 × 10³**	1.52 × 10³	1.37 × 10³	1.20 × 10³	1.52 × 10³	1.53 × 10³	1.18 × 10³	1.18 × 10³	1.31 × 10³	2.09 × 10³	1.17 × 10³
	Std	**1.54 × 10⁰**	2.86 × 10²	2.39 × 10²	1.30 × 10²	2.49 × 10²	2.87 × 10²	1.21 × 10²	1.15 × 10²	1.89 × 10²	3.47 × 10²	8.96 × 10¹
	Rank	**1**	8	7	5	9	10	3	4	6	12	2
F34	Ave	**1.46 × 10³**	2.38 × 10³	1.92 × 10³	1.71 × 10³	2.91 × 10³	1.86 × 10³	2.09 × 10³	1.70 × 10³	1.84 × 10³	2.71 × 10³	1.98 × 10³
	Std	**1.48 × 10²**	4.29 × 10²	3.44 × 10²	1.95 × 10²	5.11 × 10²	2.55 × 10²	3.30 × 10²	3.34 × 10²	3.83 × 10²	2.70 × 10²	3.58 × 10²
	Rank	**1**	9	6	3	12	5	8	2	4	10	7
F35	Ave	**1200.082**	1200.297	1200.228	1200.216	1200.796	1200.402	1200.368	1200.336	1200.810	1200.253	1200.187
	Std	**5.75 × 10⁻²**	1.70 × 10⁻¹	1.32 × 10⁻¹	1.43 × 10⁻¹	2.71 × 10⁻¹	1.72 × 10⁻¹	2.37 × 10⁻¹	2.11 × 10⁻¹	6.53 × 10⁻¹	2.95 × 10⁻¹	7.65 × 10⁻²
	Rank	**1**	4	3	5	11	9	8	7	10	6	2
F36	Ave	1300.089	1300.295	1300.294	1300.393	1300.348	1300.260	1300.457	**1300.067**	1300.193	1300.546	1300.251
	Std	**3.41 × 10⁻²**	1.17 × 10⁻¹	8.19 × 10⁻²	1.78 × 10⁻¹	1.29 × 10⁻¹	1.60 × 10⁻¹	2.37 × 10⁻¹	4.05 × 10⁻²	5.53 × 10⁻²	2.23 × 10⁻¹	8.93 × 10⁻²
	Rank	**2**	6	4	9	8	7	11	1	3	10	5
F37	Ave	**1400.141**	1400.263	1400.342	1400.395	1400.304	1400.360	1400.330	1400.173	1400.328	1400.443	1400.204
	Std	**5.51 × 10⁻²**	1.70 × 10⁻¹	1.17 × 10⁻¹	1.55 × 10⁻¹	2.21 × 10⁻¹	2.64 × 10⁻¹	3.23 × 10⁻¹	7.64 × 10⁻²	2.24 × 10⁻¹	3.02 × 10⁻¹	8.35 × 10⁻²
	Rank	**1**	4	6	8	7	9	10	2	5	11	3
F38	Ave	**1500.704**	1501.619	1501.905	1502.776	1506.984	1503.018	1503.884	1501.027	1501.553	1516.130	1501.125
	Std	**2.52 × 10⁻¹**	7.50 × 10⁻¹	8.76 × 10⁻¹	1.18 × 10⁰	2.32 × 10⁰	1.74 × 10⁰	1.80 × 10⁰	2.99 × 10⁻¹	7.46 × 10⁻¹	5.48 × 10⁰	5.27 × 10⁻¹
	Rank	**1**	4	5	8	10	7	9	2	6	12	3
F39	Ave	1601.922	1602.837	1602.933	1602.655	1603.393	1603.048	1603.024	1602.230	1602.618	1604.103	1602.640
	Std	**4.49 × 10⁻¹**	3.58 × 10⁻¹	5.45 × 10⁻¹	4.17 × 10⁻¹	3.12 × 10⁻¹	4.54 × 10⁻¹	2.99 × 10⁻¹	4.67 × 10⁻¹	5.46 × 10⁻¹	2.52 × 10⁻¹	3.33 × 10⁻¹
	Rank	**1**	8	6	5	11	7	9	2	3	12	4
Hybrid												
F40	Ave	**1.71 × 10³**	6.18 × 10³	2.33 × 10³	6.87 × 10³	2.03 × 10⁵	2.25 × 10³	8.06 × 10³	4.36 × 10³	4.08 × 10⁴	6.96 × 10³	7.91 × 10³
	Std	**5.98 × 10⁰**	3.52 × 10³	3.21 × 10²	3.05 × 10³	3.86 × 10⁵	2.75 × 10²	4.32 × 10³	2.26 × 10³	1.05 × 10⁵	1.04 × 10⁴	5.82 × 10³
	Rank	**1**	5	3	6	12	2	9	4	11	7	8
F41	Ave	**1.80 × 10³**	1.28 × 10⁴	1.96 × 10³	1.01 × 10⁴	1.19 × 10⁴	1.87 × 10³	1.10 × 10⁴	8.74 × 10³	1.23 × 10⁴	8.36 × 10³	1.88 × 10⁴
	Std	**6.62 × 10⁻¹**	1.02 × 10⁴	9.41 × 10¹	3.39 × 10³	1.18 × 10⁴	4.03 × 10¹	7.50 × 10³	5.95 × 10³	7.67 × 10³	6.04 × 10³	1.51 × 10⁴
	Rank	**1**	10	3	6	8	2	7	5	9	4	11
F42	Ave	**1900.169**	1903.329	1902.593	1903.068	1905.922	1902.772	1903.983	1901.714	1902.745	1910.302	1901.843
	Std	**1.09 × 10⁻¹**	1.05 × 10⁰	1.21 × 10⁰	9.14 × 10⁻¹	1.37 × 10⁰	1.09 × 10⁰	1.13 × 10⁰	8.04 × 10⁻¹	1.12 × 10⁰	5.64 × 10⁰	5.65 × 10⁻¹
	Rank	**1**	8	5	7	11	6	9	2	4	12	3
F43	Ave	**2.00 × 10³**	3.12 × 10³	2.08 × 10³	3.97 × 10³	8.11 × 10³	2.06 × 10³	6.96 × 10³	2.12 × 10³	7.27 × 10³	1.09 × 10⁴	5.23 × 10³
	Std	**4.55 × 10⁻¹**	1.90 × 10³	6.49 × 10¹	1.99 × 10³	4.26 × 10³	4.38 × 10¹	5.99 × 10³	6.51 × 10¹	4.46 × 10³	8.86 × 10³	4.68 × 10³
	Rank	**1**	5	3	6	11	2	9	4	10	12	7
F44	Ave	**2.10 × 10³**	5.08 × 10³	2.49 × 10³	5.06 × 10³	8.87 × 10⁴	2.46 × 10³	9.98 × 10³	2.40 × 10³	9.63 × 10³	8.09 × 10³	4.65 × 10³
	Std	**2.60 × 10⁻¹**	2.65 × 10³	2.57 × 10²	2.61 × 10³	1.56 × 10⁵	2.80 × 10²	7.23 × 10³	1.96 × 10²	4.37 × 10³	5.70 × 10³	3.86 × 10³
	Rank	**1**	7	4	6	12	3	11	2	10	8	5
F45	Ave	**2.20 × 10³**	2.26 × 10³	2.27 × 10³	2.29 × 10³	2.29 × 10³	2.24 × 10³	2.27 × 10³	2.24 × 10³	2.30 × 10³	2.49 × 10³	2.23 × 10³
	Std	**2.32 × 10⁻¹**	4.62 × 10¹	6.74 × 10¹	5.75 × 10¹	7.29 × 10¹	3.06 × 10¹	6.17 × 10¹	4.99 × 10¹	5.77 × 10¹	1.40 × 10²	2.97 × 10¹
	Rank	**1**	6	8	10	9	3	7	4	11	12	2

(*continued on next page*)

is gradually shifting from exploration to exploitation to explore most regions in the search space at the beginning of the optimization process to avoid stagnation into local minima while reducing diversity quickly during the second half of the optimization process to accelerate the convergence toward the most promising region obtained so-far.

During the last half of the optimization process, when population diversity is reduced and the exploratory process is largely transformed into the exploitative phase, the NOA convergence curves show an accelerated reducing pattern on various test functions as depicted through the third column in Fig. 16. From this column, it is clear that NOA has a slow convergence at the

Table 4 (*continued*).

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Composition												
F46	Ave	**2.50 × 10³**	2.63 × 10³	2.50 × 10³	2.50 × 10³	2.62 × 10³	2.50 × 10³	2.50 × 10³	2.63 × 10³	2.63 × 10³	2.50 × 10³	2.50 × 10³
	Std	**0.00 × 10⁰**	1.80 × 10⁻⁵	0.00 × 10⁰	0.00 × 10⁰	4.06 × 10¹	0.00 × 10⁰	0.00 × 10⁰	1.68 × 10⁻¹²	4.02 × 10⁰	0.00 × 10⁰	0.00 × 10⁰
	Rank	**1**	10	2	3	8	4	5	9	11	6	7
F47	Ave	**2.51 × 10³**	2.53 × 10³	2.54 × 10³	2.58 × 10³	2.58 × 10³	2.58 × 10³	2.58 × 10³	2.55 × 10³	2.54 × 10³	2.60 × 10³	2.55 × 10³
	Std	**5.38 × 10⁰**	1.61 × 10¹	2.01 × 10¹	3.05 × 10¹	3.14 × 10¹	3.05 × 10¹	3.01 × 10¹	3.70 × 10¹	3.46 × 10¹	1.42 × 10¹	3.16 × 10¹
	Rank	**1**	2	3	9	8	11	10	5	4	12	6
F48	Ave	**2.62 × 10³**	2.69 × 10³	2.68 × 10³	2.70 × 10³	2.69 × 10³	2.70 × 10³	2.69 × 10³	2.70 × 10³	2.70 × 10³	2.70 × 10³	2.68 × 10³
	Std	**5.89 × 10⁰**	2.81 × 10¹	2.56 × 10¹	9.51 × 10⁻¹	9.32 × 10⁰	8.53 × 10⁰	1.37 × 10¹	1.44 × 10¹	1.29 × 10¹	0.00 × 10⁰	2.58 × 10¹
	Rank	**1**	4	2	11	5	8	6	7	9	12	3
F49	Ave	2700.092	2700.202	2700.243	2700.181	2700.378	2700.229	2700.354	**2700.068**	2700.142	2700.432	2700.231
	Std	**2.48 × 10⁻²**	8.86 × 10⁻²	9.59 × 10⁻²	9.16 × 10⁻²	1.72 × 10⁻¹	1.23 × 10⁻¹	1.58 × 10⁻¹	**2.70 × 10⁻²**	1.82 × 10¹	1.50 × 10⁻¹	6.70 × 10⁻²
	Rank	2	4	7	3	10	6	9	**1**	12	8	5
F50	Ave	**2.78 × 10³**	3.01 × 10³	2.83 × 10³	2.88 × 10³	3.10 × 10³	2.84 × 10³	2.90 × 10³	3.02 × 10³	2.98 × 10³	2.90 × 10³	2.89 × 10³
	Std	**9.85 × 10¹**	1.71 × 10²	9.50 × 10¹	5.83 × 10¹	1.13 × 10²	9.34 × 10¹	0.00 × 10⁰	1.12 × 10²	1.46 × 10²	0.00 × 10⁰	5.01 × 10¹
	Rank	**1**	9	2	4	12	3	6	11	8	7	5
F51	Ave	3.02 × 10³	3.22 × 10³	**3.00 × 10³**	3.00 × 10³	3.38 × 10³	3.00 × 10³	3.00 × 10³	3.22 × 10³	3.27 × 10³	3.00 × 10³	3.00 × 10³
	Std	5.04 × 10¹	5.94 × 10¹	**0.00 × 10⁰**	0.00 × 10⁰	1.48 × 10²	0.00 × 10⁰	0.00 × 10⁰	5.14 × 10¹	7.62 × 10¹	0.00 × 10⁰	0.00 × 10⁰
	Rank	7	9	**1**	2	12	3	4	8	10	5	6
F52	Ave	**3.09 × 10³**	1.19 × 10⁵	2.69 × 10⁵	3.58 × 10³	1.77 × 10⁵	1.12 × 10⁵	3.61 × 10³	2.41 × 10⁵	3.77 × 10⁵	3.10 × 10³	3.57 × 10³
	Std	**3.64 × 10¹**	4.37 × 10⁵	6.91 × 10⁵	3.86 × 10²	5.26 × 10⁵	5.96 × 10⁵	4.55 × 10²	6.18 × 10⁵	8.60 × 10⁵	0.00 × 10⁰	3.90 × 10²
	Rank	**1**	8	11	4	9	7	5	10	12	2	3
F53	Ave	3.50 × 10³	4.42 × 10³	3.87 × 10³	4.30 × 10³	5.18 × 10³	3.98 × 10³	4.59 × 10³	3.74 × 10³	4.45 × 10³	**3.37 × 10³**	3.62 × 10³
	Std	3.10 × 10¹	6.69 × 10²	3.14 × 10²	5.31 × 10²	1.19 × 10³	4.74 × 10²	4.98 × 10²	2.94 × 10²	7.85 × 10²	**6.76 × 10²**	2.34 × 10²
	Rank	2	8	5	7	12	6	10	4	9	**1**	3
Ave.	Rank	**1.52**	6.21	4.48	6.24	9.97	5.62	7.14	4.38	8.07	9.17	4.90
Ave.	Std	**1.37 × 10¹**	2.76 × 10⁴	2.44 × 10⁴	4.70 × 10³	2.12 × 10⁵	2.02 × 10⁴	4.96 × 10³	2.32 × 10⁴	2.39 × 10⁶	1.43 × 10⁵	3.18 × 10³

Bold values indicate the best findings.

beginning of the optimization process to cover as much search space as possible to avoid getting stuck in local minima, then it is maximized in the last half of this process due to shifting from exploration to exploitation of the most promising regions obtained.

Using the same graph, the average fitness history curve shows that NOA's competitiveness among solutions to the validated test functions has decreased over time. This is because all solutions are now focused on exploiting the areas around the best-so-far solution, so the fitness values of all solutions are mostly the same. Also, Fig. 16 shows the trajectory of NOA's search for the optimal position of the first dimension as it gradually explores all aspects of the search space. To find a better solution quickly, the exploratory approach is being replaced by an exploitative one that narrows the scope of the search to one aspect. NOA's trajectory curve shows that it begins with an exploratory trend, then transitions to an exploitation one to find better outcomes before terminating the optimization process. Last but not least, the history of NOA positions is depicted in the final column of Fig. 16. As can be seen in this column, NOA does not follow the same pattern for every transfer function. For example in the case of F54, NOA first explores the entire search space before narrowing its scope near the interval of 0 and −50 in an attempt to find the optimal solution. The search history graph also shows that NOA has a more dispersed pattern for the multimodal and composition test functions, whereas its performance for the unimodal test function is more concentrated around the optimum points.

5.7. The overall effectiveness of the proposed algorithm

NOA has been evaluated separately in the previous sections using four test suites: CEC2005, CEC2014, CEC2017, and CEC2020, and compared to ten well-established metaheuristic algorithms, but the overall performance of NOA across all test suites needs to be elaborated. As a result, this section compares the overall performance of NOA and the other algorithms across all test functions of each benchmark. Table 10 shows the average rank values and standard deviations for each test suite. This table also shows the overall effectiveness of the proposed algorithm and other competing algorithms using an additional metric known as overall effectiveness (OE), which is calculated using the formula below [100]:

$$\mathbf{OE}\,(\%) = \frac{\boldsymbol{N'} - \boldsymbol{L_i}}{\boldsymbol{N'}} * \mathbf{100} \tag{21}$$

where $\boldsymbol{N'}$ denotes the number of mathematical functions in each test suite and $\boldsymbol{L_i}$ stands for the number of test functions, which NOA failed to be the best compared to all the optimizers. According to the findings in this table, NOA could be superior in terms of Std, Rank, and OE for three test suites: CEC2014, CEC2017, and CEC2020 when compared to all the rival optimizers. Unfortunately, NOA could not achieve better outcomes compared to GTO using CEC2005, but their results are slightly competitive. Overall evaluation defined in the last three rows in this table confirms that NOA is the best-performing optimizers since it outperforms all in terms of Avg. Std, Avg. Rank, and Avg. OE(%). This efficiency is due to the variation in the optimization process of NOA, where it has two various mechanisms with two distinct exploration and exploitation operators. The first mechanism occurring in the summer and fall periods starts with an exploration operator to search for the pine seeds in different regions within the search space, followed gradually by an exploitation operator to store the explored seeds in an appropriate cache. On the other hand, the second mechanism occurs in the winter and spring seasons when NOA retrieves the caches using the spatial memory strategy based on marking the hiding location of the caches using different markers, or objects at various angles. This phase is the second exploration operator simulated by NOA to find the hiding location which include the caches. Afterward, the second exploitation operator will be called to recover the stored pine seeds. Due to these various exploration and exploitation operators simulated from the nutcracker's behaviors, NOA could achieve the superior findings explained before.

5.8. Comparison with CEC-2014 winners

This section is added to demonstrate how well our proposed NOA performs in comparison to the CEC2014 winners: L-SHADE and an adaptive variant of L-SHADE (AL-SHADE). The population size for the proposed NOA in the experiments conducted herein is 100, while the maximum number of function evaluations for

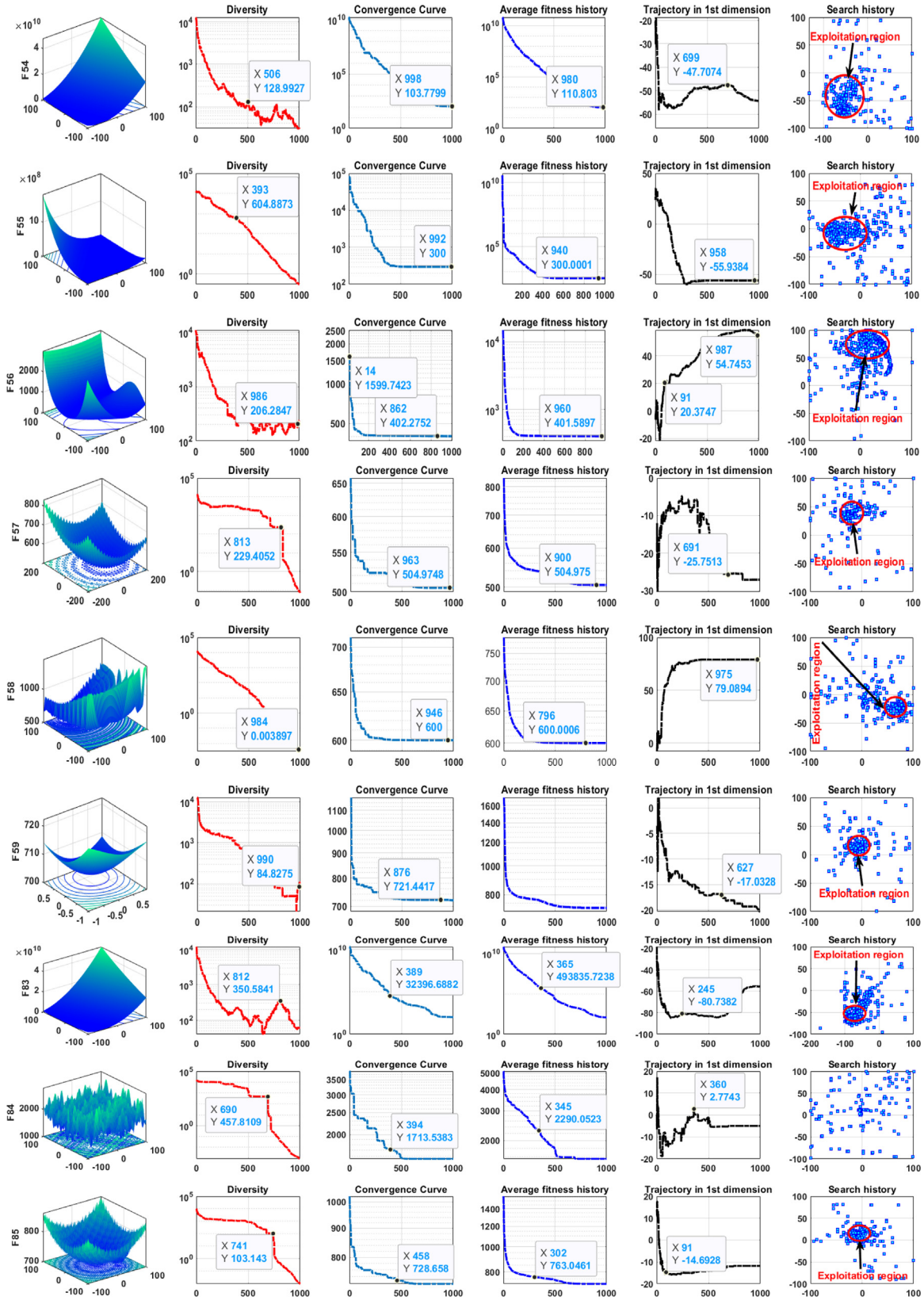


Fig. 16. Diversity, convergence curve, average fitness history, trajectory and search history.

Table 5
P-values on CEC-2014 test suite (F24–F53).

Fun	SSA		GBO		RUN		WOA		GTO		AVOA		EO		GWO		RFO		SMA	
F24	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F25	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F26	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+	**1.29 × 10^{-11}**	+
F27	**2.21 × 10^{-4}**	+	5.22 × 10^{-2}	−	1.83 × 10^{-1}	−	**3.30 × 10^{-7}**	+	**2.33 × 10^{-4}**	+	**2.20 × 10^{-5}**	+	**9.74 × 10^{-5}**	+	**2.35 × 10^{-5}**	+	**3.30 × 10^{-7}**	+	**7.17 × 10^{-7}**	+
F28	1.30 × 10^{-1}	−	4.20 × 10^{-1}	−	5.19 × 10^{-2}	−	**3.01 × 10^{-7}**	+	**1.17 × 10^{-2}**	+	**1.08 × 10^{-2}**	+	**1.20 × 10^{-10}**	+	**3.68 × 10^{-11}**	+	**9.50 × 10^{-6}**	+	**1.34 × 10^{-5}**	+
F29	**8.99 × 10^{-11}**	+	**1.09 × 10^{-10}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**1.73 × 10^{-7}**	+	**7.38 × 10^{-10}**	+	**3.02 × 10^{-11}**	+	**4.50 × 10^{-11}**	+
F30	**3.02 × 10^{-11}**	+	**5.19 × 10^{-7}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**4.62 × 10^{-10}**	+	**9.92 × 10^{-11}**	+	7.28 × 10^{-1}	−	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F31	**1.21 × 10^{-12}**	+	**1.20 × 10^{-12}**	+	**1.21 × 10^{-12}**	+	**1.21 × 10^{-12}**	+	**1.21 × 10^{-12}**	+	**1.21 × 10^{-12}**	+	**1.18 × 10^{-12}**	+	**1.21 × 10^{-12}**	+	**1.21 × 10^{-12}**	+	**1.21 × 10^{-12}**	+
F32	**4.87 × 10^{-11}**	+	**3.25 × 10^{-11}**	+	**2.96 × 10^{-11}**	+	**2.96 × 10^{-11}**	+	**5.38 × 10^{-11}**	+	**3.27 × 10^{-11}**	+	**2.43 × 10^{-7}**	+	**1.82 × 10^{-8}**	+	**2.96 × 10^{-11}**	+	**2.82 × 10^{-10}**	+
F33	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+	**2.99 × 10^{-11}**	+
F34	**1.07 × 10^{-9}**	+	**5.60 × 10^{-7}**	+	**1.76 × 10^{-1}**	+	**3.02 × 10^{-11}**	+	**3.83 × 10^{-6}**	+	**1.55 × 10^{-9}**	+	**1.44 × 10^{-3}**	+	**5.87 × 10^{-4}**	+	**3.02 × 10^{-11}**	+	**2.83 × 10^{-8}**	+
F35	**3.01 × 10^{-7}**	+	**2.84 × 10^{-4}**	+	**1.47 × 10^{-7}**	+	**3.02 × 10^{-11}**	+	**2.03 × 10^{-9}**	+	**3.20 × 10^{-9}**	+	**8.48 × 10^{-9}**	+	**2.38 × 10^{-7}**	+	**6.77 × 10^{-5}**	+	**9.21 × 10^{-5}**	+
F36	**8.99 × 10^{-11}**	+	**1.96 × 10^{-10}**	+	**3.02 × 10^{-11}**	+	**6.70 × 10^{-11}**	+	**5.07 × 10^{-10}**	+	**3.02 × 10^{-11}**	+	**1.37 × 10^{-3}**	+	**2.87 × 10^{-10}**	+	**1.33 × 10^{-10}**	+	**4.62 × 10^{-10}**	+
F37	**2.88 × 10^{-6}**	+	**8.89 × 10^{-10}**	+	**9.92 × 10^{-11}**	+	**4.69 × 10^{-8}**	+	**9.06 × 10^{-8}**	+	**1.36 × 10^{-7}**	+	5.94 × 10^{-2}	−	**2.38 × 10^{-7}**	+	**4.62 × 10^{-10}**	+	**4.64 × 10^{-3}**	+
F38	**1.11 × 10^{-6}**	+	**4.18 × 10^{-9}**	+	**9.92 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**2.61 × 10^{-10}**	+	**5.57 × 10^{-10}**	+	**8.12 × 10^{-4}**	+	**1.64 × 10^{-5}**	+	**3.02 × 10^{-11}**	+	**3.37 × 10^{-5}**	+
F39	**7.77 × 10^{-9}**	+	**1.17 × 10^{-9}**	+	**7.04 × 10^{-7}**	+	**3.02 × 10^{-11}**	+	**6.12 × 10^{-10}**	+	**5.07 × 10^{-10}**	+	**8.31 × 10^{-3}**	+	**1.87 × 10^{-5}**	+	**3.02 × 10^{-11}**	+	**9.83 × 10^{-8}**	+
F40	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F41	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F42	**3.02 × 10^{-11}**	+	**4.98 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.34 × 10^{-11}**	+
F43	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F44	**3.02 × 10^{-11}**	+	**3.34 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**5.49 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+
F45	**3.02 × 10^{-11}**	+	**8.15 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**4.50 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**2.15 × 10^{-10}**	+
F46	**1.21 × 10^{-12}**	+	NaN	±	NaN	±	**1.66 × 10^{-11}**	+	NaN	±	NaN	±	**5.99 × 10^{-13}**	+	**1.21 × 10^{-12}**	+	NaN	±	NaN	±
F47	**1.61 × 10^{-10}**	+	**5.51 × 10^{-10}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**5.73 × 10^{-10}**	+	**1.27 × 10^{-11}**	+	**8.29 × 10^{-6}**	+	**6.72 × 10^{-10}**	+	**1.21 × 10^{-12}**	+	**2.58 × 10^{-10}**	+
F48	**4.18 × 10^{-9}**	+	**3.77 × 10^{-10}**	+	**4.57 × 10^{-12}**	+	**2.95 × 10^{-10}**	+	**7.72 × 10^{-11}**	+	**1.99 × 10^{-11}**	+	**1.28 × 10^{-10}**	+	**2.61 × 10^{-10}**	+	**4.57 × 10^{-12}**	+	**2.16 × 10^{-11}**	+
F49	**7.60 × 10^{-7}**	+	**5.00 × 10^{-9}**	+	**6.53 × 10^{-8}**	+	**1.09 × 10^{-10}**	+	**1.07 × 10^{-9}**	+	**8.89 × 10^{-10}**	+	**8.12 × 10^{-4}**	+	**1.29 × 10^{-6}**	+	**3.34 × 10^{-11}**	+	**6.07 × 10^{-11}**	+
F50	**1.29 × 10^{-5}**	+	**7.03 × 10^{-4}**	+	**3.05 × 10^{-5}**	+	**8.01 × 10^{-9}**	+	**9.12 × 10^{-4}**	+	**7.88 × 10^{-7}**	+	**8.75 × 10^{-9}**	+	**9.55 × 10^{-9}**	+	**7.88 × 10^{-7}**	+	**1.79 × 10^{-6}**	+
F51	**8.16 × 10^{-12}**	+	**3.34 × 10^{-1}**	+	3.34 × 10^{-1}	−	**1.72 × 10^{-12}**	+	3.34 × 10^{-1}	−	3.34 × 10^{-1}	−	**1.93 × 10^{-12}**	+	**2.70 × 10^{-12}**	+	3.34 × 10^{-1}	−	3.34 × 10^{-1}	−
F52	**3.01 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**1.78 × 10^{-10}**	+	**3.01 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**4.99 × 10^{-9}**	+	**3.02 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**4.04 × 10^{-1}**	+	**4.99 × 10^{-9}**	+
F53	**3.02 × 10^{-11}**	+	**7.39 × 10^{-11}**	+	**8.99 × 10^{-11}**	+	**3.02 × 10^{-11}**	+	**8.10 × 10^{-10}**	+	**3.02 × 10^{-11}**	+	**3.32 × 10^{-6}**	+	**1.78 × 10^{-10}**	+	**1.24 × 10^{-9}**	+	**1.17 × 10^{-2}**	+
D	**29**		**29**		**28**		**30**		**30**		**30**		**28**		**30**		**29**		**29**	
ND	**1**		**1**		**2**		**0**		**0**		**0**		**2**		**0**		**1**		**1**	

+ stands for no difference − indicates there is the difference.
D refers to the number of test functions which contain a difference.
ND indicates the number of test functions with no difference.

NOA, L-SHADE, and AL-SHADE is 1,000,000. The other parameters of NOA are set as defined before, while those of the CEC2014 winners are set as recommended in [97,98]. Those algorithms were run 30 times independently on the CEC2014 test functions, and the results of the analysis of the fitness values in terms of Ave and Std are shown in Table 11. Inspecting this table reveals that NOA is competitive with AL-SHADE and L-SHADE for 5 test functions, superior for 14 test functions, and inferior for 11 test functions. In addition, overall effectiveness is reported in Fig. 17 which shows the average of the fitness values and Std values listed in Table 11. This figure reveals that NOA is the best in terms of the average fitness value with 1593, while its performance is slightly poor for the average Std value compared to L-SHADE. From that, it could be observed that NOA could overcome the

Table 6
Results of unimodal, multimodal, hybrid, and composition CEC-2017 test functions.

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Unimodal												
F54	Ave	**1.00 × 10²**	3.19 × 10³	1.49 × 10³	3.52 × 10³	2.10 × 10⁶	2.52 × 10³	3.28 × 10³	3.69 × 10³	1.40 × 10⁷	3.90 × 10³	6.74 × 10³
	Std	**2.75 × 10⁻⁹**	3.15 × 10³	1.92 × 10³	1.80 × 10³	6.21 × 10⁶	2.41 × 10³	3.28 × 10³	3.25 × 10³	6.39 × 10⁷	3.85 × 10³	4.64 × 10³
	Rank	**1**	4	2	6	10	3	5	7	11	8	9
F55	Ave	**3.00 × 10²**	3.00 × 10²	3.00 × 10²	3.00 × 10²	1.25 × 10³	3.00 × 10²	3.00 × 10²	3.00 × 10²	2.22 × 10³	5.70 × 10²	3.00 × 10²
	Std	**3.17 × 10⁻¹⁴**	8.02 × 10⁻¹⁰	1.24 × 10⁻¹³	9.45 × 10⁻⁴	1.00 × 10³	1.67 × 10⁻¹⁰	2.75 × 10⁻⁹	1.19 × 10⁻⁷	2.31 × 10³	7.01 × 10²	7.91 × 10⁻⁴
	Rank	**1**	4	2	8	10	3	5	6	12	9	7
Multimodal												
F56	Ave	**4.00 × 10²**	4.06 × 10²	4.01 × 10²	4.03 × 10²	4.36 × 10²	4.02 × 10²	4.06 × 10²	4.03 × 10²	4.23 × 10²	4.12 × 10²	4.21 × 10²
	Std	**1.20 × 10⁻⁵**	1.23 × 10¹	9.11 × 10⁻¹	2.27 × 10⁰	4.10 × 10¹	1.09 × 10⁰	1.16 × 10¹	1.57 × 10⁰	2.14 × 10¹	1.90 × 10¹	2.85 × 10¹
	Rank	**1**	6	2	4	11	3	7	5	10	8	9
F57	Ave	**5.05 × 10²**	5.21 × 10²	5.25 × 10²	5.30 × 10²	5.52 × 10²	5.27 × 10²	5.38 × 10²	5.13 × 10²	5.19 × 10²	5.59 × 10²	5.17 × 10²
	Std	**2.21 × 10⁰**	8.98 × 10⁰	1.06 × 10¹	8.56 × 10⁰	1.94 × 10¹	1.10 × 10¹	1.74 × 10¹	5.89 × 10⁰	8.64 × 10⁰	2.16 × 10¹	5.59 × 10⁰
	Rank	**1**	5	6	8	11	7	9	2	4	12	3
F58	Ave	**600.000**	609.029	601.412	617.617	629.777	607.177	609.314	600.010	601.180	643.471	600.220
	Std	**1.52 × 10⁻⁴**	6.84 × 10⁰	1.08 × 10⁰	7.23 × 10⁰	1.37 × 10¹	4.29 × 10⁰	1.46 × 10¹	7.84 × 10⁻³	1.23 × 10⁰	1.02 × 10¹	2.63 × 10⁻²
	Rank	**1**	7	4	8	11	6	9	2	5	12	3
F59	Ave	**7.16 × 10²**	7.34 × 10²	7.39 × 10²	7.58 × 10²	7.79 × 10²	7.49 × 10²	7.63 × 10²	7.27 × 10²	7.32 × 10²	8.15 × 10²	7.26 × 10²
	Std	**3.00 × 10⁰**	1.21 × 10¹	1.56 × 10¹	1.48 × 10¹	2.41 × 10¹	1.69 × 10¹	1.66 × 10¹	1.06 × 10¹	8.35 × 10⁰	6.78 × 10⁰	5.67 × 10⁰
	Rank	**1**	5	6	8	11	7	9	3	4	12	2
F60	Ave	**8.05 × 10²**	8.22 × 10²	8.23 × 10²	8.27 × 10²	8.43 × 10²	8.25 × 10²	8.29 × 10²	8.12 × 10²	8.13 × 10²	8.52 × 10²	8.17 × 10²
	Std	**2.70 × 10⁰**	8.87 × 10⁰	8.20 × 10⁰	5.17 × 10⁰	1.75 × 10¹	8.36 × 10⁰	1.17 × 10¹	4.98 × 10⁰	5.57 × 10⁰	1.78 × 10¹	7.37 × 10⁰
	Rank	**1**	5	6	8	10	7	9	2	3	12	4
F61	Ave	**900.000**	919.841	910.349	1063.171	1406.620	993.623	1155.089	900.624	922.635	1749.439	900.004
	Std	**2.99 × 10⁻¹⁴**	5.17 × 10¹	1.23 × 10¹	9.60 × 10¹	3.89 × 10²	7.59 × 10¹	1.78 × 10²	5.26 × 10⁻¹	2.36 × 10¹	1.47 × 10²	8.43 × 10⁻²
	Rank	**1**	6	4	9	11	7	10	3	5	12	2
F62	Ave	**1.14 × 10³**	1.83 × 10³	1.84 × 10³	1.66 × 10³	2.05 × 10³	1.90 × 10³	1.84 × 10³	1.53 × 10³	1.55 × 10³	2.34 × 10³	1.55 × 10³
	Std	**1.03 × 10²**	2.92 × 10²	2.97 × 10²	1.97 × 10²	3.52 × 10²	3.15 × 10²	3.70 × 10²	2.24 × 10²	2.44 × 10²	3.26 × 10²	1.96 × 10²
	Rank	**1**	6	8	5	10	9	7	2	4	11	3
Hybrid												
F63	Ave	**1.10 × 10³**	2.05 × 10³	1.11 × 10³	1.13 × 10³	1.24 × 10³	1.12 × 10³	1.14 × 10³	1.11 × 10³	1.37 × 10³	9.61 × 10⁴	1.11 × 10³
	Std	**1.43 × 10⁰**	5.27 × 10²	5.28 × 10⁰	8.04 × 10⁰	1.66 × 10²	1.42 × 10¹	2.63 × 10¹	8.34 × 10⁰	9.10 × 10²	1.58 × 10⁵	7.28 × 10⁰
	Rank	**1**	11	3	6	8	5	7	2	10	12	4
F64	Ave	**1.23 × 10³**	1.71 × 10⁶	1.52 × 10⁴	6.92 × 10⁴	4.78 × 10⁶	1.81 × 10⁴	4.29 × 10⁵	9.21 × 10³	8.37 × 10⁵	3.51 × 10⁵	6.55 × 10⁴
	Std	**3.41 × 10¹**	2.24 × 10⁶	1.34 × 10⁴	1.00 × 10⁵	5.38 × 10⁶	1.54 × 10⁴	4.81 × 10⁵	7.43 × 10³	1.52 × 10⁶	3.97 × 10⁵	5.73 × 10⁴
	Rank	**1**	10	3	6	11	4	8	2	9	7	5
F65	Ave	**1.30 × 10³**	1.45 × 10⁴	1.82 × 10³	9.12 × 10³	1.72 × 10⁴	1.59 × 10³	1.35 × 10⁴	6.72 × 10³	1.09 × 10⁴	1.58 × 10⁴	1.16 × 10⁴
	Std	**2.41 × 10⁰**	1.32 × 10⁴	5.44 × 10²	4.52 × 10³	1.40 × 10⁴	2.66 × 10²	9.18 × 10³	5.84 × 10³	6.75 × 10³	1.41 × 10⁴	1.24 × 10⁴
	Rank	**1**	9	3	5	11	2	8	4	6	10	7
F66	Ave	**1.40 × 10³**	1.50 × 10³	1.48 × 10³	1.95 × 10³	2.45 × 10³	1.46 × 10³	1.99 × 10³	1.46 × 10³	3.33 × 10³	4.66 × 10³	1.50 × 10³
	Std	**6.58 × 10⁻¹**	3.43 × 10¹	3.61 × 10¹	5.70 × 10²	1.29 × 10³	2.91 × 10¹	7.56 × 10²	2.11 × 10¹	1.79 × 10³	3.49 × 10³	2.14 × 10²
	Rank	**1**	5	4	8	10	2	9	3	11	12	6
F67	Ave	**1.50 × 10³**	2.50 × 10³	1.57 × 10³	2.04 × 10³	6.26 × 10³	1.56 × 10³	3.48 × 10³	1.57 × 10³	4.68 × 10³	1.48 × 10⁴	1.94 × 10³
	Std	**4.77 × 10⁻¹**	7.37 × 10²	4.55 × 10¹	6.43 × 10²	4.15 × 10³	5.72 × 10¹	1.29 × 10³	4.35 × 10¹	3.66 × 10³	1.15 × 10⁴	8.55 × 10²
	Rank	**1**	8	3	6	11	2	9	4	10	12	5
F68	Ave	**1.61 × 10³**	1.72 × 10³	1.75 × 10³	1.79 × 10³	1.88 × 10³	1.69 × 10³	1.78 × 10³	1.68 × 10³	1.79 × 10³	2.16 × 10³	1.70 × 10³
	Std	**3.00 × 10¹**	1.07 × 10²	1.15 × 10²	1.27 × 10²	1.18 × 10²	9.93 × 10¹	1.33 × 10²	7.81 × 10¹	1.43 × 10²	2.09 × 10²	6.94 × 10¹
	Rank	**1**	6	7	9	11	3	8	2	10	12	4
F69	Ave	**1.70 × 10³**	1.78 × 10³	1.75 × 10³	1.76 × 10³	1.81 × 10³	1.75 × 10³	1.76 × 10³	1.74 × 10³	1.76 × 10³	1.91 × 10³	1.76 × 10³
	Std	**9.27 × 10⁻¹**	4.08 × 10¹	3.66 × 10¹	1.44 × 10¹	4.84 × 10¹	1.70 × 10¹	3.55 × 10¹	1.67 × 10¹	3.60 × 10¹	1.58 × 10²	4.27 × 10¹
	Rank	**1**	9	4	6	11	3	8	2	5	12	7
F70	Ave	**1.80 × 10³**	1.23 × 10⁴	3.71 × 10³	1.63 × 10⁴	1.75 × 10⁴	2.63 × 10³	2.03 × 10⁴	1.39 × 10⁴	2.46 × 10⁴	1.72 × 10⁴	2.70 × 10⁴
	Std	**2.74 × 10⁻¹**	9.40 × 10³	3.31 × 10³	1.03 × 10⁴	1.38 × 10⁴	1.61 × 10³	1.16 × 10⁴	1.25 × 10⁴	1.40 × 10⁴	1.13 × 10⁴	1.60 × 10⁴
	Rank	**1**	4	3	6	8	2	9	5	10	7	11
F71	Ave	**1.90 × 10³**	2.26 × 10³	1.98 × 10³	7.50 × 10³	7.49 × 10⁴	1.96 × 10³	5.56 × 10³	1.95 × 10³	1.56 × 10⁴	8.27 × 10³	6.16 × 10³
	Std	**3.08 × 10⁻²**	5.61 × 10²	6.74 × 10¹	6.29 × 10³	1.78 × 10⁵	5.35 × 10¹	4.68 × 10³	3.42 × 10¹	4.13 × 10⁴	6.22 × 10³	6.44 × 10³
	Rank	**1**	5	4	9	12	3	7	2	11	10	8
F72	Ave	**2.00 × 10³**	2.10 × 10³	2.08 × 10³	2.12 × 10³	2.17 × 10³	2.07 × 10³	2.11 × 10³	2.06 × 10³	2.08 × 10³	2.25 × 10³	2.03 × 10³
	Std	**2.31 × 10⁰**	6.35 × 10¹	5.69 × 10¹	5.83 × 10¹	5.93 × 10¹	4.68 × 10¹	6.05 × 10¹	5.70 × 10¹	5.44 × 10¹	1.03 × 10²	2.67 × 10¹
	Rank	**1**	8	5	10	11	4	9	3	6	12	2

(*continued on next page*)

CEC2014 winners for the majority of the CEC2014 test functions. Hence, it could be classified as a highly-performing optimizer.

5.9. Comparison with CEC2017 winners

This new section compares our suggested NOA's performance to that of the CEC2017 winners, LSHADE-SPACMA and LSHADE-cnEpSin. The experiments conducted here were based on a population size of 200 for the proposed NOA, and a total of 1,000,000 function evaluations for NOA, LSHADE-SPACMA and LSHADE-cnEpSin. The other NOA parameters are set as previously defined, while the CEC2017 winners' parameters are set as suggested in [92,99]. The results of the analysis of the fitness values in terms of Ave and Std are displayed in Table 12 after those algorithms were applied 30 times independently to the CEC2017 test functions. Looking at this table reveals that NOA is superior for 18 test functions, inferior for four test functions, and competitive LSHADE-SPACMA and LSHADE-cnEpSin for 7 test functions. Additionally, Fig. 18 (which displays the average of the fitness values and Std values listed in Table 11) reports overall effectiveness. This figure shows that NOA is the best in terms of average fitness value with 1677, but that it performs slightly worse than LSHADE-SPACMA for average Std value. This affirms our conclusion that NOA could be categorized as a highly-performing optimizer.

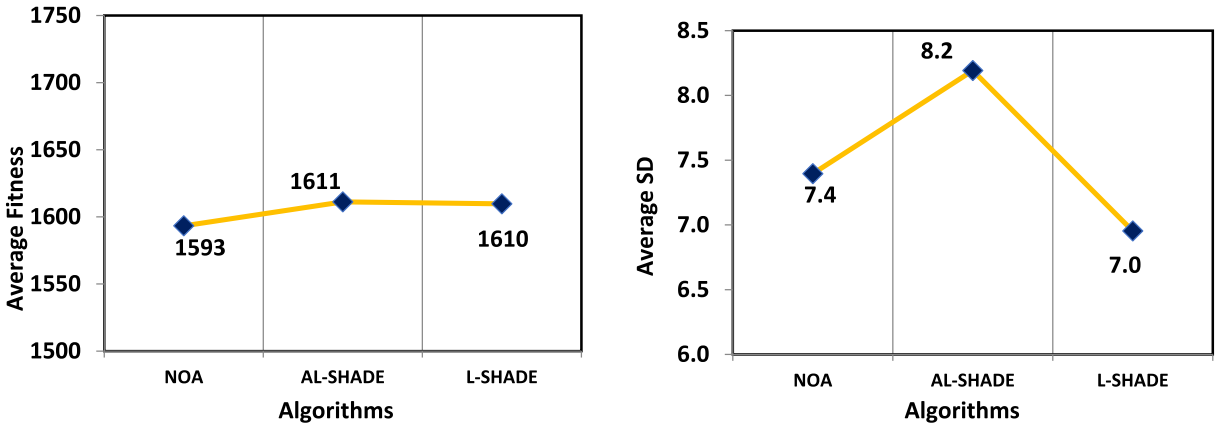
6. Real-world optimization problems

This section talks about the experiments that were done to see how well the NOA worked with five engineering problems:

Table 6 (*continued*).

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Composition												
F73	Ave	2.25×10^{3}	2.28×10^{3}	2.27×10^{3}	2.22×10^{3}	2.31×10^{3}	$\mathbf{2.22 \times 10^{3}}$	2.26×10^{3}	2.30×10^{3}	2.30×10^{3}	2.35×10^{3}	2.30×10^{3}
	Std	5.43×10^{1}	5.82×10^{1}	6.27×10^{1}	4.49×10^{1}	6.99×10^{1}	$\mathbf{4.51 \times 10^{1}}$	6.79×10^{1}	3.47×10^{1}	3.46×10^{1}	4.37×10^{1}	4.91×10^{1}
	Rank	4	7	6	2	11	**1**	5	9	10	12	8
F74	Ave	$\mathbf{2.27 \times 10^{3}}$	2.30×10^{3}	2.30×10^{3}	2.31×10^{3}	2.44×10^{3}	2.30×10^{3}	2.31×10^{3}	2.30×10^{3}	2.31×10^{3}	2.38×10^{3}	2.34×10^{3}
	Std	$\mathbf{4.49 \times 10^{1}}$	2.31×10^{1}	1.26×10^{0}	5.13×10^{0}	3.58×10^{2}	1.24×10^{1}	9.07×10^{0}	1.13×10^{1}	1.97×10^{1}	9.00×10^{1}	1.69×10^{2}
	Rank	**1**	2	4	7	12	5	6	3	8	11	9
F75	Ave	$\mathbf{2.61 \times 10^{3}}$	2.62×10^{3}	2.63×10^{3}	2.62×10^{3}	2.65×10^{3}	2.63×10^{3}	2.64×10^{3}	2.61×10^{3}	2.62×10^{3}	2.72×10^{3}	2.62×10^{3}
	Std	$\mathbf{3.21 \times 10^{0}}$	9.22×10^{0}	1.35×10^{1}	1.07×10^{1}	2.73×10^{1}	1.39×10^{1}	1.76×10^{1}	4.86×10^{0}	9.09×10^{0}	4.07×10^{1}	5.69×10^{0}
	Rank	**1**	5	8	6	10	7	9	2	3	12	4
F76	Ave	$\mathbf{2.62 \times 10^{3}}$	2.74×10^{3}	2.73×10^{3}	2.75×10^{3}	2.78×10^{3}	2.74×10^{3}	2.78×10^{3}	2.74×10^{3}	2.75×10^{3}	2.81×10^{3}	2.76×10^{3}
	Std	$\mathbf{1.19 \times 10^{2}}$	4.61×10^{1}	7.98×10^{1}	7.63×10^{0}	2.72×10^{1}	6.47×10^{1}	2.33×10^{1}	5.54×10^{0}	1.09×10^{1}	1.31×10^{2}	1.05×10^{1}
	Rank	**1**	4	2	6	10	3	9	5	7	12	8
F77	Ave	**2914.927**	2919.836	2917.628	2924.635	2952.137	2916.308	2925.348	2926.734	2932.432	2946.625	2931.421
	Std	$\mathbf{2.30 \times 10^{1}}$	2.34×10^{1}	2.28×10^{1}	2.41×10^{1}	6.15×10^{1}	2.41×10^{1}	3.54×10^{1}	2.00×10^{1}	1.61×10^{1}	6.92×10^{1}	2.64×10^{1}
	Rank	**1**	4	9	2	10	3	11	8	7	5	6
F78	Ave	$\mathbf{2.85 \times 10^{3}}$	2.97×10^{3}	3.05×10^{3}	3.02×10^{3}	3.60×10^{3}	2.96×10^{3}	3.08×10^{3}	3.03×10^{3}	3.09×10^{3}	3.81×10^{3}	3.30×10^{3}
	Std	$\mathbf{1.04 \times 10^{2}}$	2.81×10^{2}	2.60×10^{2}	2.56×10^{2}	6.43×10^{2}	1.41×10^{2}	2.99×10^{2}	2.95×10^{2}	3.30×10^{2}	5.08×10^{2}	5.00×10^{2}
	Rank	**1**	3	6	4	11	2	7	5	9	12	10
F79	Ave	3.09×10^{3}	3.09×10^{3}	3.10×10^{3}	3.09×10^{3}	3.14×10^{3}	3.10×10^{3}	3.10×10^{3}	3.09×10^{3}	3.10×10^{3}	3.19×10^{3}	$\mathbf{3.09 \times 10^{3}}$
	Std	2.20×10^{0}	2.91×10^{0}	1.84×10^{1}	2.22×10^{0}	4.02×10^{1}	8.44×10^{0}	1.18×10^{1}	3.09×10^{0}	1.28×10^{1}	4.32×10^{1}	$\mathbf{1.69 \times 10^{0}}$
	Rank	2	3	10	5	11	6	8	4	7	12	**1**
F80	Ave	$\mathbf{3.12 \times 10^{3}}$	3.28×10^{3}	3.33×10^{3}	3.34×10^{3}	3.40×10^{3}	3.31×10^{3}	3.31×10^{3}	3.35×10^{3}	3.40×10^{3}	3.39×10^{3}	3.30×10^{3}
	Std	$\mathbf{7.91 \times 10^{1}}$	1.87×10^{2}	1.30×10^{2}	1.27×10^{2}	2.02×10^{2}	1.39×10^{2}	1.36×10^{2}	1.14×10^{2}	1.10×10^{2}	1.22×10^{2}	1.48×10^{2}
	Rank	**1**	2	7	8	12	6	5	9	11	10	4
F81	Ave	$\mathbf{3.14 \times 10^{3}}$	3.19×10^{3}	3.24×10^{3}	3.19×10^{3}	3.33×10^{3}	3.23×10^{3}	3.27×10^{3}	3.18×10^{3}	3.20×10^{3}	3.49×10^{3}	3.22×10^{3}
	Std	$\mathbf{8.16 \times 10^{0}}$	5.28×10^{1}	5.78×10^{1}	3.08×10^{1}	9.27×10^{1}	6.50×10^{1}	7.01×10^{1}	3.18×10^{1}	4.83×10^{1}	1.53×10^{2}	7.02×10^{1}
	Rank	**1**	3	9	4	11	8	10	2	5	12	6
F82	Ave	$\mathbf{3.47 \times 10^{3}}$	2.38×10^{5}	3.88×10^{5}	1.06×10^{5}	9.61×10^{5}	2.02×10^{6}	1.77×10^{5}	3.94×10^{5}	6.04×10^{5}	1.86×10^{6}	1.70×10^{5}
	Std	$\mathbf{8.74 \times 10^{1}}$	4.51×10^{5}	5.70×10^{5}	1.62×10^{5}	1.13×10^{6}	3.08×10^{6}	2.61×10^{5}	5.29×10^{5}	7.52×10^{5}	2.20×10^{6}	3.79×10^{5}
	Rank	**1**	5	6	2	10	12	4	7	8	11	3
Ave. Rank		**1.14**	5.57	5.00	6.39	10.61	4.39	7.93	3.86	7.61	10.71	5.61
Ave. Std		$\mathbf{2.44 \times 10^{1}}$	9.38×10^{4}	2.04×10^{4}	9.90×10^{3}	4.46×10^{5}	1.07×10^{5}	2.67×10^{4}	1.93×10^{4}	2.28×10^{6}	9.68×10^{4}	1.65×10^{4}

Bold values indicate the best findings.



a) Average of fitness values of all test functions b) Average of Std values of all test functions

Fig. 17. Comparison among L-SHADE, AL-SHADE, and NOA using CEC2014.

a welded beam, a tension/compression spring, a pressure vessel, a 3-bar truss, and a 10-bar truss. It is not easy to solve these kinds of problems due to finding some constraints which have to be handled. There are a variety of penalty methods that can be used to deal with these constraints, some of the more common ones are static, dynamic, annealing, adaptive, co-evolutionary, and death penalty [101]. With NOA and rival optimizers, the death penalty is used to give the unfeasible solutions an objective value high enough to be discarded during the optimization process if minimization is desired.

6.1. The welded beam design problem

In this section, NOA's performance is evaluated using the welded bean design (WBD) problem depicted in Fig. 19 (a). To minimize the total fabrication cost of a welded beam, it is necessary to determine the optimal values for four design variables (h, l, t, and b) to satisfy seven constraints, including tip detection, weld coverage, bending stress, buckling load, and cost. Following is a description of the WBD problem's mathematical model:

– Solutions to this problem are represented as follows:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} = \begin{bmatrix} h & l & t & b \end{bmatrix}$$

– The objective function employed along with the optimization algorithm to handle this problem is as:

$$f(X) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

– The mathematical formulation of the constraints of this problem:

$$g_1(X) = \tau(X) - \tau_{max} \leq 0$$
$$g_2(X) = \sigma(X) - \sigma_{max} \leq 0$$
$$g_3(X) = \delta(X) - \delta_{max} \leq 0$$

Table 7
P-values on CEC-2017 test suite (F54–F82).

Fun	SSA		GBO		RUN		WOA		GTO		AVOA		EO		GWO		RFO		SMA	
F54	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F55	1.72 × 10^{-12}	+	9.24 × 10^{-12}	+	1.72 × 10^{-12}	+	1.72 × 10^{-12}	+	1.80 × 10^{-12}	+	1.72 × 10^{-12}	+	1.80 × 10^{-12}	+	1.72 × 10^{-12}	+	1.72 × 10^{-12}	+	1.72 × 10^{-12}	+
F56	3.02 × 10^{-11}	+	2.37 × 10^{-10}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F57	1.29 × 10^{-10}	+	3.58 × 10^{-11}	−	2.94 × 10^{-11}	−	2.94 × 10^{-11}	+	3.59 × 10^{-11}	+	2.93 × 10^{-11}	−	4.09 × 10^{-10}	+	1.73 × 10^{-10}	+	2.94 × 10^{-11}	+	9.52 × 10^{-10}	+
F58	2.11 × 10^{-11}	+	2.11 × 10^{-11}	−	2.11 × 10^{-11}	+	2.11 × 10^{-11}	+	2.11 × 10^{-11}	+	2.11 × 10^{-11}	−	2.04 × 10^{-6}	+	2.11 × 10^{-11}	+	2.11 × 10^{-11}	+	2.11 × 10^{-11}	+
F59	8.10 × 10^{-10}	+	3.34 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	1.07 × 10^{-7}	+	5.49 × 10^{-11}	+	3.02 × 10^{-11}	+	1.41 × 10^{-9}	+
F60	1.04 × 10^{-10}	+	1.04 × 10^{-10}	+	2.87 × 10^{-11}	+	2.87 × 10^{-11}	+	3.88 × 10^{-11}	+	3.51 × 10^{-11}	+	3.03 × 10^{-7}	−	1.94 × 10^{-8}	+	2.87 × 10^{-11}	+	6.31 × 10^{-8}	+
F61	1.21 × 10^{-12}	+	1.20 × 10^{-12}	+	1.21 × 10^{-12}	+	1.21 × 10^{-12}	+	1.21 × 10^{-12}	+	1.21 × 10^{-12}	+	1.64 × 10^{-10}	+	1.21 × 10^{-12}	+	1.21 × 10^{-12}	+	1.21 × 10^{-12}	+
F62	7.39 × 10^{-11}	+	1.33 × 10^{-10}	+	3.69 × 10^{-11}	+	6.07 × 10^{-11}	+	6.69 × 10^{-11}	+	3.69 × 10^{-11}	+	1.55 × 10^{-9}	+	1.86 × 10^{-9}	+	3.02 × 10^{-11}	+	5.49 × 10^{-11}	+
F63	3.02 × 10^{-11}	+	1.46 × 10^{-10}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	4.98 × 10^{-11}	+	3.69 × 10^{-11}	+	3.35 × 10^{-8}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	4.98 × 10^{-11}	+
F64	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F65	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F66	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F67	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F68	7.39 × 10^{-11}	+	5.07 × 10^{-10}	+	5.49 × 10^{-11}	+	4.98 × 10^{-11}	+	2.15 × 10^{-10}	+	6.07 × 10^{-11}	+	2.15 × 10^{-10}	+	1.21 × 10^{-10}	+	3.02 × 10^{-11}	+	1.33 × 10^{-10}	+
F69	3.02 × 10^{-11}	+	9.92 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	4.08 × 10^{-11}	+	3.02 × 10^{-11}	+	4.62 × 10^{-10}	+
F70	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F71	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
F72	2.62 × 10^{-11}	+	2.89 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+	2.62 × 10^{-11}	+
F73	1.34 × 10^{-5}	+	1.60 × 10^{-4}	+	2.09 × 10^{-3}	+	1.10 × 10^{-6}	+	1.59 × 10^{-2}	+	1.42 × 10^{-4}	+	8.95 × 10^{-8}	+	8.62 × 10^{-10}	+	6.51 × 10^{-10}	+	1.29 × 10^{-8}	+
F74	1.59 × 10^{-10}	+	1.19 × 10^{-10}	+	2.98 × 10^{-11}	+	1.84 × 10^{-9}	+	1.31 × 10^{-10}	+	2.98 × 10^{-11}	+	1.78 × 10^{-4}	+	9.80 × 10^{-11}	+	2.98 × 10^{-11}	+	1.42 × 10^{-8}	+
F75	1.85 × 10^{-8}	+	5.57 × 10^{-10}	+	3.47 × 10^{-10}	+	7.39 × 10^{-11}	+	4.57 × 10^{-9}	+	3.02 × 10^{-11}	+	1.49 × 10^{-6}	+	2.57 × 10^{-7}	+	3.02 × 10^{-11}	+	5.57 × 10^{-10}	+
F76	1.04 × 10^{-9}	+	9.92 × 10^{-9}	±	1.26 × 10^{-9}	±	3.43 × 10^{-9}	+	5.29 × 10^{-6}	±	7.19 × 10^{-10}	±	7.64 × 10^{-9}	+	4.90 × 10^{-9}	+	2.63 × 10^{-11}	±	8.70 × 10^{-11}	+
F77	5.26 × 10^{-3}	+	6.26 × 10^{-4}	+	5.83 × 10^{-5}	+	1.36 × 10^{-8}	+	1.85 × 10^{-4}	+	7.00 × 10^{-6}	+	4.37 × 10^{-4}	+	1.94 × 10^{-5}	+	1.25 × 10^{-8}	+	2.48 × 10^{-4}	+
F78	2.98 × 10^{-9}	+	9.64 × 10^{-7}	+	4.31 × 10^{-2}	+	1.96 × 10^{-11}	+	4.54 × 10^{-7}	+	7.89 × 10^{-5}	+	7.61 × 10^{-9}	+	1.96 × 10^{-11}	+	2.60 × 10^{-10}	+	2.60 × 10^{-10}	+
F79	3.51 × 10^{-2}	+	8.25 × 10^{-6}	+	9.50 × 10^{-4}	+	4.47 × 10^{-11}	+	2.38 × 10^{-4}	+	5.06 × 10^{-8}	+	1.45 × 10^{-1}	+	6.71 × 10^{-6}	+	3.00 × 10^{-11}	+	8.10 × 10^{-4}	+
F80	5.12 × 10^{-3}	+	8.29 × 10^{-3}	+	2.59 × 10^{-6}	+	5.24 × 10^{-8}	+	1.12 × 10^{-4}	+	1.83 × 10^{-5}	+	1.85 × 10^{-4}	+	8.09 × 10^{-6}	+	3.77 × 10^{-5}	+	1.94 × 10^{-6}	+
F81	4.44 × 10^{-7}	+	6.70 × 10^{-11}	+	2.57 × 10^{-7}	+	4.98 × 10^{-11}	+	1.47 × 10^{-7}	+	2.37 × 10^{-10}	+	7.60 × 10^{-7}	+	1.41 × 10^{-9}	+	3.02 × 10^{-11}	+	3.57 × 10^{-6}	+
F82	3.02 × 10^{-11}	+	3.34 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	4.96 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+	3.02 × 10^{-11}	+
D	29		26		27		29		28		27		29		29		29		29	
ND	0		3		2		0		1		2		0		0		0		0	

+ stands for no difference − indicates there is the difference.
D refers to the number of the test functions which contain a difference.
ND indicates the number of test functions with no difference.

$$g_4(X) = x_1 - x_4 \le 0$$
$$g_5(X) = P - P_c(X) \le 0$$
$$g_6(X) = 0.125 - x_1 \le 0$$
$$g_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$$

Where

$$\tau(\mathrm{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ \mathrm{M} = \mathrm{P}\left(\mathrm{L} + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

Table 8
Results of unimodal, multimodal, hybrid, and composition CEC-2020 test functions.

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Unimodal												
F83	Ave	**1.00 × 10²**	2.19 × 10³	2.10 × 10³	2.90 × 10³	3.05 × 10⁶	2.12 × 10³	3.75 × 10³	3.19 × 10³	4.89 × 10⁷	3.61 × 10³	7.44 × 10³
	Std	**2.17 × 10⁻⁹**	2.58 × 10³	2.29 × 10³	1.13 × 10³	7.30 × 10⁶	2.22 × 10³	3.45 × 10³	2.86 × 10³	1.40 × 10⁸	3.16 × 10³	4.37 × 10³
	Rank	**1**	4	2	5	10	3	8	6	11	7	9
Multimodal												
F84	Ave	**1.23 × 10³**	1.81 × 10³	1.90 × 10³	1.55 × 10³	2.12 × 10³	1.93 × 10³	1.91 × 10³	1.52 × 10³	1.54 × 10³	2.45 × 10³	1.64 × 10³
	Std	**9.56 × 10¹**	2.84 × 10²	3.00 × 10²	2.26 × 10²	3.39 × 10²	2.51 × 10²	3.30 × 10²	2.11 × 10²	1.94 × 10²	4.17 × 10²	1.85 × 10²
	Rank	**1**	6	7	4	10	9	8	2	3	12	5
Hybrid												
F85	Ave	**7.16 × 10²**	7.32 × 10²	7.38 × 10²	7.61 × 10²	7.78 × 10²	7.48 × 10²	7.59 × 10²	7.21 × 10²	7.30 × 10²	8.14 × 10²	7.27 × 10²
	Std	**2.84 × 10⁰**	9.69 × 10⁰	1.17 × 10¹	1.62 × 10¹	2.11 × 10¹	1.61 × 10¹	1.69 × 10¹	4.48 × 10⁰	8.20 × 10⁰	6.00 × 10⁰	8.58 × 10⁰
	Rank	**1**	5	6	9	11	7	8	2	4	12	3
F86	Ave	**1900.000**	1901.608	**1900.000**	**1900.000**	1900.050	**1900.000**	**1900.000**	**1900.000**	1900.373	**1900.000**	**1900.000**
	Std	**0.00 × 10⁰**	7.24 × 10⁻¹	**0.00 × 10⁰**	**0.00 × 10⁰**	1.99 × 10⁻¹	**0.00 × 10⁰**	**0.00 × 10⁰**	**0.00 × 10⁰**	5.44 × 10⁻¹	**0.00 × 10⁰**	**0.00 × 10⁰**
	Rank	**1**	5	**1**	**1**	2	**1**	**1**	**1**	3	**1**	**1**
F87	Ave	**1.71 × 10³**	5.88 × 10³	2.41 × 10³	1.16 × 10⁴	3.40 × 10⁵	2.23 × 10³	8.49 × 10³	3.90 × 10³	5.24 × 10⁴	2.88 × 10⁴	7.82 × 10³
	Std	**1.01 × 10¹**	3.38 × 10³	3.68 × 10²	2.85 × 10⁴	6.45 × 10⁵	3.24 × 10²	4.72 × 10³	1.83 × 10³	1.22 × 10⁵	6.18 × 10⁴	5.48 × 10³
	Rank	**1**	5	3	8	12	2	7	4	11	10	6
F88	Ave	**1.60 × 10³**	1.72 × 10³	1.73 × 10³	1.72 × 10³	1.86 × 10³	1.76 × 10³	1.77 × 10³	1.69 × 10³	1.75 × 10³	2.03 × 10³	1.71 × 10³
	Std	**3.41 × 10⁰**	7.76 × 10¹	8.15 × 10¹	8.40 × 10¹	1.04 × 10²	8.58 × 10¹	1.09 × 10²	8.49 × 10¹	1.10 × 10²	2.18 × 10²	8.99 × 10¹
	Rank	**1**	5	6	4	11	8	9	2	7	12	3
F89	Ave	**2.10 × 10³**	6.82 × 10³	2.45 × 10³	4.47 × 10³	5.52 × 10⁴	2.41 × 10³	8.01 × 10³	2.34 × 10³	1.03 × 10⁴	8.45 × 10³	4.09 × 10³
	Std	**3.06 × 10⁰**	4.46 × 10³	2.72 × 10²	2.15 × 10³	5.44 × 10⁴	1.89 × 10²	6.33 × 10³	1.42 × 10²	4.62 × 10³	6.50 × 10³	3.79 × 10³
	Rank	**1**	7	4	6	12	3	8	2	11	9	5
Composition												
F90	Ave	**2.27 × 10³**	2.32 × 10³	2.30 × 10³	2.30 × 10³	2.32 × 10³	2.31 × 10³	2.30 × 10³	2.31 × 10³	2.36 × 10³	2.39 × 10³	2.35 × 10³
	Std	**4.48 × 10¹**	1.40 × 10²	7.68 × 10⁰	1.29 × 10¹	2.17 × 10¹	3.51 × 10⁰	1.65 × 10¹	7.74 × 10¹	1.61 × 10²	2.17 × 10²	2.15 × 10²
	Rank	**1**	8	2	4	7	5	3	6	11	12	9
F91	Ave	**2.59 × 10³**	2.74 × 10³	2.72 × 10³	2.75 × 10³	2.76 × 10³	2.69 × 10³	2.74 × 10³	2.74 × 10³	2.74 × 10³	2.86 × 10³	2.75 × 10³
	Std	**1.14 × 10²**	4.61 × 10¹	8.84 × 10¹	9.96 × 10⁰	6.95 × 10¹	1.16 × 10²	9.68 × 10¹	4.78 × 10⁰	1.14 × 10¹	5.65 × 10¹	4.77 × 10¹
	Rank	**1**	4	3	9	10	2	5	6	7	12	8
F92	Ave	**2.92 × 10³**	2.93 × 10³	2.93 × 10³	2.92 × 10³	2.95 × 10³	2.93 × 10³	2.94 × 10³	2.93 × 10³	2.94 × 10³	2.95 × 10³	2.93 × 10³
	Std	**2.24 × 10¹**	2.39 × 10¹	5.43 × 10¹	2.50 × 10¹	6.73 × 10¹	2.49 × 10¹	2.92 × 10¹	2.15 × 10¹	1.70 × 10¹	3.44 × 10¹	3.25 × 10¹
	Rank	**1**	4	7	2	10	3	8	5	9	11	6
Ave Rank		**1.00**	6.11	4.05	5.58	9.68	4.79	6.95	4.37	8.47	10.37	6.21
Ave Std		**2.82 × 10¹**	1.05 × 10³	3.31 × 10²	3.06 × 10³	7.62 × 10⁵	3.08 × 10²	1.44 × 10³	4.99 × 10²	1.33 × 10⁷	6.90 × 10³	1.35 × 10³

Bold values indicate the best findings.

Table 9
P-values on CEC-2020 test suite (F83–F92).

Fun	SSA		GBO		RUN		WOA		GTO		AVOA		EO		GWO		RFO		SMA	
F83	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+
F84	**1.09 × 10⁻¹⁰**	+	**2.67 × 10⁻⁹**	+	**1.01 × 10⁻⁸**	+	**9.92 × 10⁻¹¹**	+	**2.61 × 10⁻¹⁰**	+	**3.02 × 10⁻¹¹**	+	**1.73 × 10⁻⁴**	+	**2.67 × 10⁻⁹**	+	**3.02 × 10⁻¹¹**	+	**1.31 × 10⁻⁸**	+
F85	**3.02 × 10⁻¹¹**	+	**6.07 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**1.61 × 10⁻⁶**	+	**5.07 × 10⁻¹⁰**	+	**3.02 × 10⁻¹¹**	+	**2.03 × 10⁻⁹**	+
F86	**1.21 × 10⁻¹²**	+	NaN	±	NaN	−	8.15 × 10⁻⁰²	+	NaN	±	NaN	±	NaN	±	**5.85 × 10⁻⁰⁹**	+	NaN	±	NaN	±
F87	**3.02 × 10⁻¹¹**	−	**3.34 × 10⁻¹¹**	−	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**4.50 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+
F88	**3.69 × 10⁻¹¹**	+	**4.08 × 10⁻¹¹**	+	**4.50 × 10⁻¹¹**	+	**3.69 × 10⁻¹¹**	+	**5.49 × 10⁻¹¹**	+	**4.98 × 10⁻¹¹**	+	**8.99 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.69 × 10⁻¹¹**	+
F89	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.34 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+	**3.02 × 10⁻¹¹**	+
F90	**2.14 × 10⁻¹⁰**	+	**1.69 × 10⁻⁹**	+	**1.95 × 10⁻¹⁰**	+	**3.00 × 10⁻¹¹**	+	**3.00 × 10⁻¹¹**	+	**6.66 × 10⁻¹¹**	+	**1.78 × 10⁻⁴**	+	**5.54 × 10⁻¹⁰**	+	**6.49 × 10⁻⁹**	+	**1.06 × 10⁻⁹**	+
F91	**8.95 × 10⁻¹⁰**	+	**2.53 × 10⁻⁹**	+	**8.95 × 10⁻¹⁰**	+	**1.30 × 10⁻⁹**	+	**2.48 × 10⁻⁸**	+	**8.95 × 10⁻¹⁰**	+	**4.13 × 10⁻⁸**	+	**2.72 × 10⁻⁷**	+	**8.95 × 10⁻¹⁰**	+	**8.85 × 10⁻¹¹**	+
F92	**8.12 × 10⁻³**	+	**2.42 × 10⁻³**	+	1.44 × 10⁻¹	−	**4.79 × 10⁻⁷**	+	**3.61 × 10⁻⁴**	+	**1.42 × 10⁻⁴**	+	**4.28 × 10⁻⁴**	+	**1.80 × 10⁻²**	+	**4.12 × 10⁻⁵**	+	**3.58 × 10⁻⁶**	+
D	**10**		**9**		8		**10**		**9**		**9**		**9**		**9**		**9**		**9**	
ND	**0**		**1**		2		**0**		**1**		**1**		**1**		**1**		**1**		**1**	

+ stands for no difference − indicates there is difference.
D refers to the number of the test functions which contain a difference.
ND indicates the number of test functions with no difference.

$$\sigma(X) = \frac{6PL}{x_4 x_3^2},\ \delta(X) = \frac{6PL^3}{E x_3^2 x_4},$$

$$P_c(X) = \frac{4.0134E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000, L = 14, E = 30 \times 10^6, G = 12 \times 10^6,$$

$$\tau_{max} = 13600, \sigma_{max} = 30000, \delta_{max} = 0.25$$

Table 13 presents the results obtained by NOA and these of 10 optimizers on this problem, which demonstrates that NOA is superior to all optimizers in terms of average fitness value (Ave), Std, Rank, and *p*-value, and competitive in terms of the best fitness value. A further illustration is provided in Fig. 19(b), which depicts the convergence speed obtained by the proposed algorithm on the WBD problem; this figure shows that NOA has a

Table 10
Overall effectiveness of the proposed NOA.

F	Index	NOA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Standard	Rank	2.13	4.87	2.22	2.52	3.70	**1.35**	1.30	3.13	4.43	3.61	2.09
	Std	190.46	133.12	50.68	78.53	1335.05	**0.00**	3.64	25.99	31.02	7.31	0.06
	Oe (%)	73.91	21.74	69.57	65.22	52.17	**82.61**	78.26	43.48	34.78	47.83	60.87
CeC2014	Rank	**1.52**	6.21	4.48	6.24	9.97	5.62	7.14	4.38	8.07	9.17	4.90
	Std	**1.3 × 10¹**	2.7 × 10⁴	2.4 × 10⁴	4.7 × 10³	2.1 × 10⁵	2.0 × 10⁴	4.9 × 10³	2.3 × 10⁴	2.3 × 10⁶	1.4 × 10⁵	3.1 × 10³
	Oe (%)	**83.33**	0.00	3.33	3.33	0.00	0.00	0.00	6.67	0.00	3.33	0.00
CeC2017	Rank	**1.14**	5.57	5.00	6.39	10.61	4.39	7.93	3.86	7.61	10.71	5.61
	Std	**2.4 × 10¹**	9.3 × 10⁴	2.0 × 10⁴	9.9 × 10³	4.4 × 10⁵	1.0 × 10⁵	2.6 × 10⁴	1.9 × 10⁴	2.2 × 10⁶	9.6 × 10⁴	1.6 × 10⁴
	Oe (%)	**86.67**	0.00	0.00	0.00	0.00	3.33	0.00	0.00	0.00	0.00	3.33
CeC2020	Rank	**1.00**	6.11	4.05	5.58	9.68	4.79	6.95	4.37	8.47	10.37	6.21
	Std	**2.8 × 10¹**	1.0 × 10³	3.3 × 10²	3.0 × 10³	7.6 × 10⁵	3.0 × 10²	1.4 × 10³	4.9 × 10²	1.3 × 10⁷	6.9 × 10³	1.3 × 10³
	Oe (%)	**100.00**	0.00	6.67	6.67	0.00	6.67	6.67	6.67	0.00	6.67	6.67
Ave. Rank:		**1.4475**	5.6900	3.9375	5.1825	8.4900	4.0375	5.8300	3.9350	7.1450	8.4650	4.7025
Ave. Std:		**6.4 × 10¹**	3.0 × 10⁴	1.1 × 10⁴	4.4 × 10³	3.5 × 10⁵	3.1 × 10⁴	8.2 × 10³	1.0 × 10⁴	4.4 × 10⁶	6.1 × 10⁴	5.2 × 10³
Ave.Oe (%):		**85.98**	5.44	19.89	18.81	13.04	23.15	21.23	14.21	8.70	14.46	17.72 f

Bold values indicates the best outcomes.

Table 11
Comparison among NOA and CEC2014 winners.

F	Index	NOA	AL-SHADE	L-SHADE	F	Index	NOA	AL-SHADE	L-SHADE
F24	Ave	**100**	**100**	**100**	F39	Ave	1600.44932	**1600.30747**	1600.67692
	Std	**0**	**0**	**0**		Std	0.30035	**0.16971**	0.25434
F25	Ave	**200**	**200**	**200**	F40	Ave	**1700.00969**	1700.51814	1700.42418
	Std	**0**	**0**	**0**		Std	**0.04369**	0.56960	0.47000
F26	Ave	**300**	**300**	**300**	F41	Ave	**1800.00124**	1800.01463	1800.01068
	Std	**0**	**0**	**0**		Std	**0.00093**	0.02673	0.03965
F27	Ave	**402.31545**	432.60610	425.79456	F42	Ave	1900.01823	1900.01589	**1900.00065**
	Std	**6.43051**	8.29363	15.18790		Std	0.01273	0.01287	**0.00355**
F28	Ave	513.32592	508.64706	**505.10388**	F43	Ave	**2000.00185**	2000.06649	2000.01095
	Std	9.58395	10.05783	**8.42065**		Std	**0.00461**	0.12697	0.02398
F29	Ave	**600**	**600**	**600**	F44	Ave	**2100.00030**	2100.20595	2100.02677
	Std	**0**	**0**	**0**		Std	**0.00024**	0.20761	0.09343
F30	Ave	700.00238	700.00164	**700.00025**	F45	Ave	2200.09381	2200.00244	**2200.00143**
	Std	0.00412	0.00503	**0.00135**		Std	0.14899	0.00438	**0.00228**
F31	Ave	**800**	**800**	**800**	F46	Ave	**2500**	2629.45747	2629.45747
	Std	**0**	**0**	**0**		Std	**0**	0.00000	0.00000
F32	Ave	900.83014	901.12762	**900.79808**	F47	Ave	**2501.39420**	2504.82607	2502.48032
	Std	0.81477	1.40060	**0.67135**		Std	**2.58383**	3.61669	2.95969
F33	Ave	**1000**	**1000**	1000.00208	F48	Ave	**2604.30097**	2627.83465	2606.39256
	Std	**0**	**0**	0.01140		Std	**5.61332**	44.56134	18.66491
F34	Ave	1179.57625	**1109.64066**	1114.30168	F49	Ave	2700.03609	**2700.02398**	2700.04195
	Std	81.00588	**8.95484**	12.71759		Std	0.00855	**0.00776**	0.01452
F35	Ave	**1200.000**	1200.00170	1200.02985	F50	Ave	**2707.40183**	2747.44644	2724.34286
	Std	**1.E−07**	0.00776	0.00947		Std	**36.37667**	122.15005	89.51233
F36	Ave	1300.03648	**1300.02501**	1300.03529	F51	Ave	**3000**	3186.91015	3196.15966
	Std	0.01049	**0.00937**	0.01114		Std	**0**	37.96491	49.59883
F37	Ave	**1400.03632**	1400.03871	1400.06008	F52	Ave	**3000**	3121.99881	3121.98497
	Std	**0.01313**	0.00980	0.01647		Std	**0**	0.43931	0.46143
F38	Ave	1500.42469	**1500.30238**	1500.32914	F53	Ave	**3389.59324**	3463.71280	3465.05059
	Std	0.11235	**0.06007**	0.05471		Std	**78.87741**	7.10066	9.39030

Bold values indicates the best outcomes.

high convergence speed in the right direction of the near-optimal solution.

6.2. Tension/compression spring design problem

Fig. 20(a) shows a tension/compression spring, and the goal of this problem is to reduce its weight [102–104]. Constraints, such as shear stress, surge frequency, and deflection must be met by the optimum design. Design variables include the mean coil diameter (D), wire diameter (d), and the number of active coils (N) that must be accurately estimated to achieve the best possible design. The mathematical model of this problem is as follows:

– Solutions to this problem are represented as follows:

$$X = \left[x_1\ x_2\ x_3\right] = [d\ D\ N]$$

– The objective function employed along with the optimization algorithm to handle this problem is as:

$$\text{Minimize } f(X) = (x_3 + 2)\, x_2 x_1^2$$

– Subject to the following constraints:

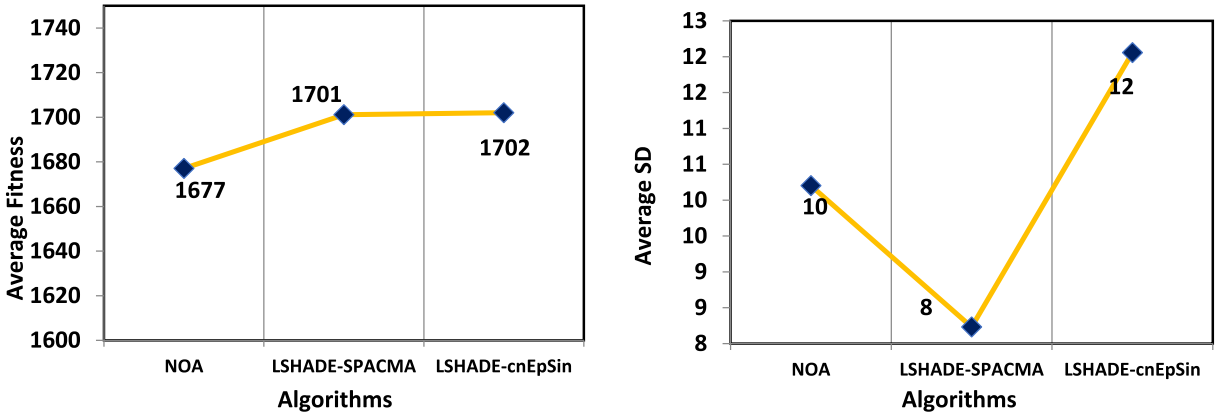
$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} \le 0$$

Table 12
Comparison among NOA and CEC2017 winners.

F	Index	NOA	LSHADE-SPACMA	LSHADE-cnEpSin	F	Index	NOA	LSHADE-SPACMA	LSHADE-cnEpSin
F54	Ave	**100**	**100**	**100**	F69	Ave	1.70049×10^{3}	$\mathbf{1.70005 \times 10^{3}}$	1.70008×10^{3}
	Std	**0**	**0**	**0**		Std	5.94648×10^{-1}	$\mathbf{9.81856 \times 10^{-2}}$	1.01036×10^{-1}
F55	Ave	**300**	**300**	**300**	F70	Ave	$\mathbf{1.80000 \times 10^{3}}$	1.80430×10^{3}	1.80234×10^{3}
	Std	**0**	**0**	**0**		Std	$\mathbf{3.15192 \times 10^{-4}}$	8.40591×10^{0}	6.38288×10^{0}
F56	Ave	**400**	**400**	**400**	F71	Ave	$\mathbf{1.90000 \times 10^{3}}$	1.90007×10^{3}	1.90015×10^{3}
	Std	**0**	**0**	**0**		Std	$\mathbf{1.82605 \times 10^{-10}}$	5.09233×10^{-2}	4.73689×10^{-1}
F57	Ave	$\mathbf{5.00895 \times 10^{2}}$	5.01193×10^{2}	5.01293×10^{2}	F72	Ave	**2000**	2.00628×10^{3}	2.00009×10^{3}
	Std	$\mathbf{7.55030 \times 10^{-1}}$	6.29267×10^{-1}	6.71546×10^{-1}		Std	**0**	9.61098×10^{0}	1.50794×10^{-1}
F58	Ave	**600**	**600**	**600**	F73	Ave	$\mathbf{2.19667 \times 10^{3}}$	2.20000×10^{3}	2.25147×10^{3}
	Std	**0**	**0**	**0**		Std	$\mathbf{1.82574 \times 10^{1}}$	3.03165×10^{-13}	5.42618×10^{1}
F59	Ave	$\mathbf{7.10958 \times 10^{2}}$	7.10997×10^{2}	7.11484×10^{2}	F74	Ave	**2200**	2.30003×10^{3}	2.30000×10^{3}
	Std	4.45408×10^{-1}	$\mathbf{3.63872 \times 10^{-1}}$	2.76966×10^{-1}		Std	**0**	9.05603×10^{-2}	2.14370×10^{-13}
F60	Ave	8.00895×10^{2}	$\mathbf{8.00795 \times 10^{2}}$	8.01794×10^{2}	F75	Ave	$\mathbf{2.60052 \times 10^{3}}$	2.60113×10^{3}	2.60055×10^{3}
	Std	9.18237×10^{-1}	$\mathbf{6.29267 \times 10^{-1}}$	7.81368×10^{-1}		Std	$\mathbf{1.19591 \times 10^{0}}$	1.45951×10^{0}	1.15288×10^{0}
F61	Ave	**900**	**900**	**900**	F76	Ave	**2500**	2.70693×10^{3}	2.68309×10^{3}
	Std	**0**	**0**	**0**		Std	$\mathbf{8.44445 \times 10^{-14}}$	7.27150×10^{1}	9.65095×10^{1}
F62	Ave	1.01866×10^{3}	$\mathbf{1.00762 \times 10^{3}}$	1.01732×10^{3}	F77	Ave	$\mathbf{2.89774 \times 10^{3}}$	2.92522×10^{3}	2.92561×10^{3}
	Std	3.07842×10^{1}	$\mathbf{6.43158 \times 10^{0}}$	3.70358×10^{1}		Std	**0**	2.34039×10^{1}	2.34325×10^{1}
F63	Ave	**1100**	**1100**	**1100**	F78	Ave	$\mathbf{2.73333 \times 10^{3}}$	2.90000×10^{3}	2.90000×10^{3}
	Std	**0**	**0**	**0**		Std	$\mathbf{1.32179 \times 10^{2}}$	0.00000×10^{0}	0
F64	Ave	$\mathbf{1.20285 \times 10^{3}}$	1.32160×10^{3}	1.30166×10^{3}	F79	Ave	$\mathbf{3.08783 \times 10^{3}}$	3.08952×10^{3}	3.08909×10^{3}
	Std	$\mathbf{2.01085 \times 10^{0}}$	7.71133×10^{1}	5.40511×10^{1}		Std	$\mathbf{1.13544 \times 10^{0}}$	4.79346×10^{-13}	1.59582×10^{0}
F65	Ave	$\mathbf{1.30049 \times 10^{3}}$	1.30297×10^{3}	1.30339×10^{3}	F80	Ave	$\mathbf{3.06000 \times 10^{3}}$	3.10000×10^{3}	3.11698×10^{3}
	Std	$\mathbf{4.98054 \times 10^{-1}}$	2.56495×10^{0}	2.33656×10^{0}		Std	1.03724×10^{2}	$\mathbf{0.00000 \times 10^{0}}$	3.57940×10^{1}
F66	Ave	**1400**	1.40029×10^{3}	**1400**	F81	Ave	3.12805×10^{3}	$\mathbf{3.12754 \times 10^{3}}$	3.12732×10^{3}
	Std	**0**	6.71546×10^{-1}	**0**		Std	3.07041×10^{0}	$\mathbf{8.63059 \times 10^{-1}}$	1.02056×10^{0}
F67	Ave	$\mathbf{1.5000 \times 10^{3}}$	1.50035×10^{3}	1.50020×10^{3}	F82	Ave	$\mathbf{3.39538 \times 10^{3}}$	3.42503×10^{3}	3.41551×10^{3}
	Std	$\mathbf{8.76352 \times 10^{-4}}$	2.06751×10^{-1}	1.74925×10^{-1}		Std	$\mathbf{2.76054 \times 10^{-1}}$	3.31098×10^{1}	3.24314×10^{1}
F68	Ave	$\mathbf{1.60002 \times 10^{3}}$	1.60066×10^{3}	1.60058×10^{3}					
	Std	$\mathbf{1.86852 \times 10^{-2}}$	2.96332×10^{-1}	4.73527×10^{-1}					

Bold values indicates the best outcomes.



a) Average of fitness values of all test functions b) Average of Std values of all test functions

Fig. 18. Comparison among LSHADE-SPACMA, LSHADE-cnEpSin, and NOA using CEC2017.

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

Variable range $0.05 \leq x_1 \leq 2.00$

$0.25 \leq x_2 \leq 1.30$

$2.00 \leq x_3 \leq 15.0$

The near-optimal value and weight of three design variables, along with some other statistical data such as Ave, Rank, *p*-value, and Std obtained by NOA and 10 rival algorithms, are found in Table 14. This table demonstrates that NOA is the most effective method due to its higher Best, Ave, Std, and Rank. Here, the proposed algorithm's convergence rate is shown on this problem in Fig. 20(b) which shows that NOA has a high rate of convergence in the right direction of the near-optimal solution.

6.3. Pressure vessel design problem

In this problem, the goal is to reduce total production by finding the best possible values for four design variables, while

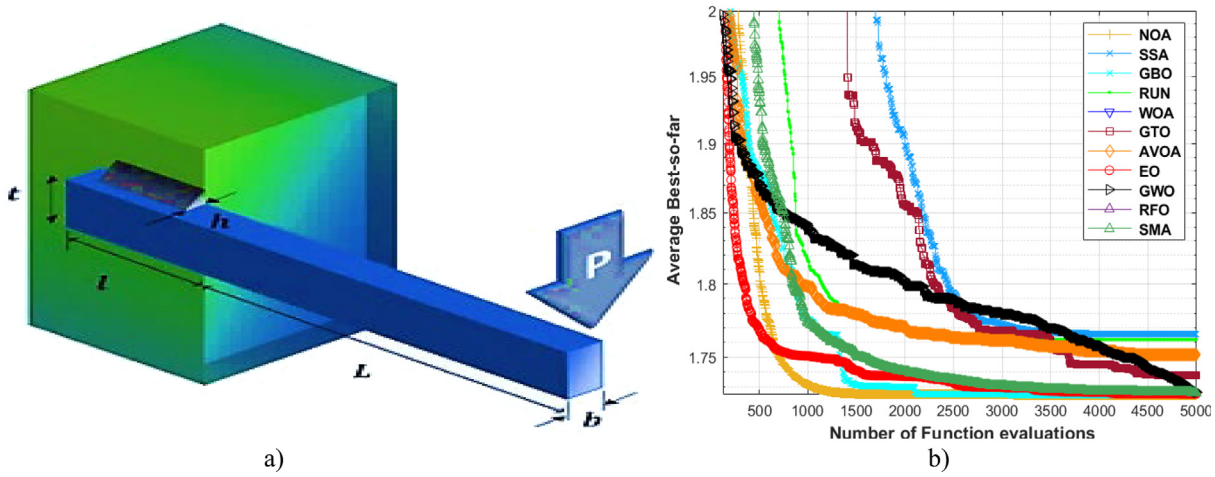


Fig. 19. The welded beam design: (a) Structure [105] and (b) Convergence curve.

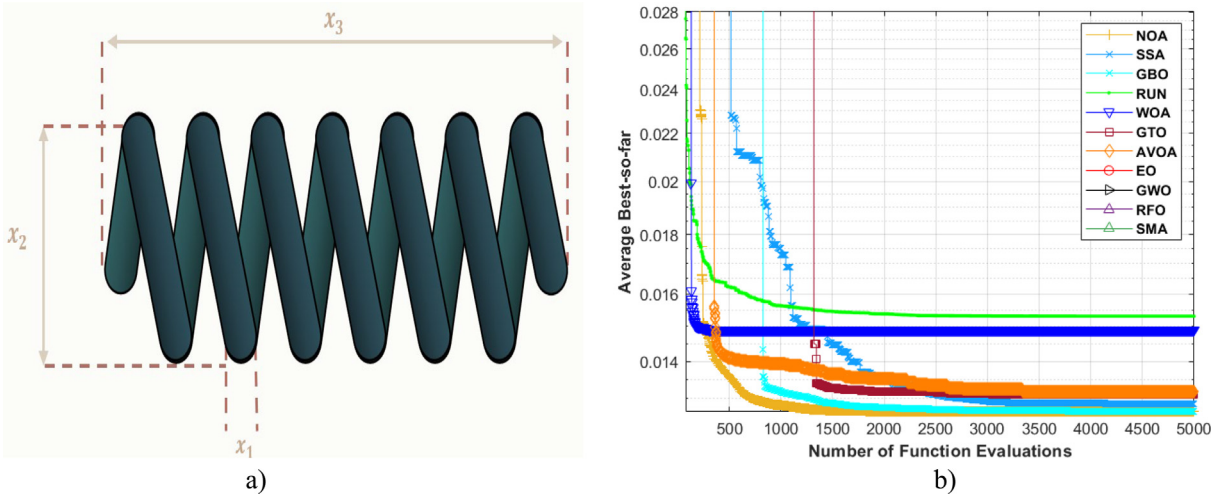


Fig. 20. Tension/compression spring design: (a) Schematic, and (b) Convergence curve.

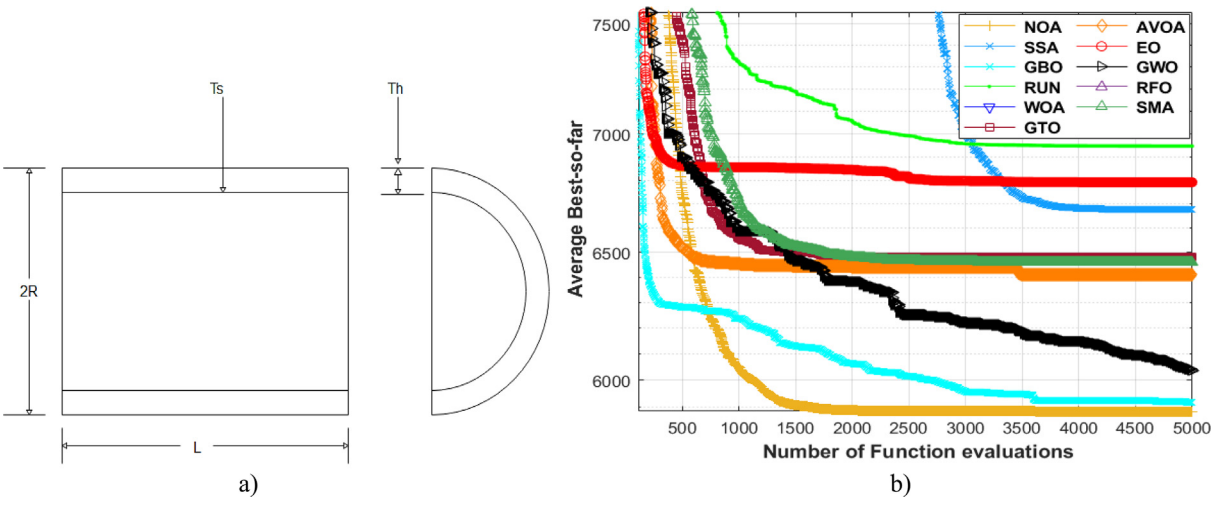


Fig. 21. Pressure Vessel: (a) Structure and (b) Convergence curve.

Table 13
Results of Welded beam design problem.

Algorithms	x_1	x_2	x_3	x_3	Best	Ave	Std	Rank	p-value
NOA	**0.20572**	**3.47070**	**9.03662**	**0.20573**	**1.72487**	**1.724866**	**4.8974 × 10^{-13}**	**1**	**3.020 × 10^{-11}**
SCA	0.19609	3.76215	9.07493	0.21069	1.79372	1.852441	3.7095 × 10^{-2}	10	3.020 × 10^{-11}
SSA	0.20438	3.49987	9.03663	0.20573	1.72671	1.782344	5.1727 × 10^{-2}	9	3.338 × 10^{-3}
GBO	**0.20572**	**3.47070**	**9.03662**	**0.20573**	**1.72487**	**1.724866**	1.6928 × 10^{-6}	2	3.020 × 10^{-11}
RUN	0.20572	3.47069	9.03660	0.20573	1.72487	1.754908	4.3029 × 10^{-2}	8	3.020 × 10^{-11}
WOA	0.17104	4.22706	9.43259	0.20383	1.82262	2.307445	5.4806 × 10^{-1}	12	8.891 × 10^{-10}
GTO	0.20572	3.47070	9.03662	0.20573	1.72487	1.750119	9.5687 × 10^{-2}	7	3.020 × 10^{-11}
AVOA	0.20570	3.47110	9.03660	0.20573	1.72490	1.742485	3.1015 × 10^{-2}	6	3.020 × 10^{-11}
EO	0.20572	3.47070	9.03662	0.20573	1.72487	1.725409	1.3164 × 10^{-3}	3	3.020 × 10^{-11}
GWO	0.20553	3.47534	9.03664	0.20574	1.72530	1.726201	8.5639 × 10^{-4}	5	3.020 × 10^{-11}
RFO	0.20572	3.47068	9.03692	0.20573	1.72490	2.076155	2.7209 × 10^{-1}	11	3.020 × 10^{-11}
SMA	0.20572	3.47070	9.03662	0.20573	1.72487	1.725904	1.6644 × 10^{-3}	4	3.020 × 10^{-11}

Table 14
Results of the Tension/compression spring design problem.

Algorithms	x_1	x_2	x_3	Best	Ave	Std	Rank	p-value
NOA	**0.05169**	**0.35671**	**11.28932**	**0.0126652**	**0.01267**	**1.5745 × 10^{-7}**	**1**	**3.020 × 10^{-11}**
SCA	0.05000	0.31738	14.06135	0.0127438	0.01295	1.2210 × 10^{-4}	3	3.020 × 10^{-11}
SSA	0.05188	0.36143	11.01809	0.0126659	0.01301	6.9395 × 10^{-4}	4	3.020 × 10^{-11}
GBO	0.05187	0.36105	11.03921	0.0126658	0.01268	1.3720 × 10^{-5}	2	3.020 × 10^{-11}
RUN	0.05034	0.32523	13.40527	0.0126991	0.01526	1.9227 × 10^{-3}	7	3.020 × 10^{-11}
WOA	0.05332	0.39732	9.25282	0.0127127	0.01458	1.3984 × 10^{-3}	6	5.427 × 10^{-11}
GTO	0.05182	0.35990	11.10471	0.0126655	16.67775	3.7901 × 10^{1}	9	3.020 × 10^{-11}
AVOA	0.05217	0.36831	10.64036	0.0126694	0.01330	8.0423 × 10^{-4}	5	2.114 × 10^{-11}
EO	0.05200	0.36428	10.85909	0.0126670	50.00788	5.0849 × 10^{1}	12	3.001 × 10^{-11}
GWO	0.05183	0.36016	11.09745	0.0126742	13.34469	3.4571 × 10^{1}	8	2.800 × 10^{-11}
RFO	0.05286	0.38547	9.78294	0.0126898	30.00981	4.6604 × 10^{1}	10	2.800 × 10^{-11}
SMA	0.05169	0.35671	11.28932	0.0126652	30.00997	4.6604 × 10^{1}	11	3.020 × 10^{-11}

Table 15
Results for pressure vessel design problem.

Algorithms	x_1	x_2	x_3	x_3	Best	Ave	Std	Rank	p-value
NOA	**0.77818**	**0.38466**	**40.31962**	**200.00000**	**5885.43417**	**5885.4342**	**1.0757 × 10^{-7}**	**1**	**3.0199 × 10^{-11}**
SCA	0.83218	0.44206	41.84167	181.80835	6289.77823	6833.4042	4.6598 × 10^{2}	9	3.0199 × 10^{-11}
SSA	0.81117	0.40097	42.02888	177.51179	5944.52042	6471.3539	4.5254 × 10^{2}	5	3.0199 × 10^{-11}
GBO	0.77818	0.38466	40.31962	200.00000	5885.43418	5976.3079	1.3660 × 10^{2}	3	3.0199 × 10^{-11}
RUN	0.77818	0.38466	40.31971	199.99884	5885.45929	6992.9746	5.6795 × 10^{2}	10	3.0199 × 10^{-11}
WOA	0.78146	0.50567	40.32326	199.94933	6258.58401	7982.7928	1.4796 × 10^{3}	11	5.5605 × 10^{-10}
GTO	0.77818	0.38466	40.31962	200.00000	5885.43417	6575.6347	6.1669 × 10^{2}	6	3.0199 × 10^{-11}
AVOA	0.77834	0.38474	40.32804	199.88287	5885.71191	6661.2247	6.0691 × 10^{2}	7	3.0123 × 10^{-11}
EO	0.77972	0.38542	40.39936	198.89289	5888.07006	6712.2988	5.2585 × 10^{2}	8	3.0199 × 10^{-11}
GWO	0.77840	0.38496	40.32522	199.94274	5887.61463	5942.6436	2.2099 × 10^{2}	2	3.0199 × 10^{-11}
RFO	0.79256	0.39185	41.06460	189.91741	5911.50327	57488.2073	6.1630 × 10^{4}	12	3.0199 × 10^{-11}
SMA	0.77818	0.38466	40.31962	200.00000	5885.43417	6447.8593	5.3272 × 10^{2}	4	3.0199 × 10^{-11}

Table 16
Results for the three-bar truss design problem.

Algorithms	x_1	x_2	Best	Ave	Std	Rank
NOA	**0.78868**	**0.40825**	**263.89584338**	**263.89584**	**3.1667 × 10^{-14}**	**1**
SCA	0.79059	0.40291	263.90317968	265.23281	4.7872 × 10^{0}	12
SSA	0.78869	0.40820	263.89584364	263.89622	5.3089 × 10^{-4}	6
GBO	**0.78868**	**0.40825**	**263.89584338**	**263.89584**	9.4191 × 10^{-8}	2
RUN	0.78877	0.40798	263.89585142	263.91749	3.1806 × 10^{-2}	10
WOA	0.78858	0.40852	263.89585061	264.74883	1.4018 × 10^{0}	11
GTO	**0.78868**	**0.40825**	**263.89584338**	**263.89584**	1.8279 × 10^{-6}	3
AVOA	0.78882	0.40784	263.89585871	263.90183	6.9221 × 10^{-3}	9
EO	0.78871	0.40816	263.89584408	263.89594	1.1355 × 10^{-4}	4
GWO	0.78879	0.40792	263.89592072	263.89766	1.8777 × 10^{-3}	8
RFO	0.78860	0.40846	263.89584947	263.89610	3.0510 × 10^{-4}	5
SMA	**0.78868**	**0.40825**	**263.89584338**	263.89662	1.6458 × 10^{-3}	7

keeping pressure requirements in mind as an optimization constraint. The inner radius (R), the thickness of the head (Th), the thickness of the shell (Ts), and the cylindrical portion length (L) are all design variables. In Fig. 21 (a), you can see the pressure vessel's structure. The mathematical model of this problem is as follows [24]:

Table 17
Results for 10-bar truss design problem.

Algorithms	NOA	SCA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
x_1	**30.5214**	28.6123	30.9661	30.6167	30.7498	30.5638	30.5017	31.5595	30.7545	30.4202	3.49×10^{-9}	2.44×10^{-9}
x_2	**0.1000**	1.914351	0.10000	0.10000	0.10000	0.13135	0.10000	0.10000	0.10001	0.10075	0.10075	0.10000
x_3	**23.1688**	23.657021	23.9930	22.9636	23.2853	22.0098	23.3533	23.1417	23.1536	23.3338	23.3338	23.1846
x_4	**15.2421**	14.415974	14.8518	15.6008	14.0370	13.5376	14.9793	14.8764	15.2100	15.3604	15.3604	15.0983
x_5	**0.1000**	0.141659	0.1000	0.1000	0.1013	0.1313	0.1001	0.1000	0.1000	0.1026	0.1026	0.1000
x_6	**0.5534**	0.436889	0.5075	0.5617	0.3971	0.1310	0.4709	0.4958	0.5403	0.4250	0.4250	0.5390
x_7	**21.0460**	23.206193	20.9116	20.8822	21.5571	22.0132	21.0565	20.8951	20.9686	21.0624	21.0624	21.1584
x_8	**7.4607**	8.391798	7.4046	7.4580	7.5265	9.5364	7.4477	7.4879	7.4618	7.4445	7.4445	7.4829
x_9	**0.1000**	0.186644	0.1000	0.1000	0.1000	0.1313	0.1000	0.1000	0.1000	0.1001	0.1001	0.1000
x_{10}	**21.5224**	21.250641	21.1613	21.5169	21.7863	22.0792	21.6664	21.2666	21.4787	21.5265	21.5265	21.3691
Best	**5060.8594**	5190.53325	5062.7407	5061.3352	5067.0943	5131.1998	5061.5516	5062.6786	5060.9583	5063.1240	5063.1240	5061.0336
Ave	**5061.9494**	5473.525475	5072.5080	5072.3065	5077.0518	6435.9602	5132.1667	5073.4743	5067.2202	5080.9003	5069.4373	5065.3404
Std	**4.00×10^{0}**	576.7138176	8.9489	9.2100	3.1148	816.7081	207.0000	7.7513	7.7630	10.4567	6.4320	6.6346
Rank	**1**	11	6	5	8	12	10	7	3	9	4	2
p-value	**3.019×10^{-11}**	2.4386×10^{-9}	3.822×10^{-10}	5.491×10^{-11}	3.019×10^{-11}	1.464×10^{-10}	5.572×10^{-10}	1.411×10^{-9}	1.776×10^{-10}	3.471×10^{-9}	2.436×10^{-9}	3.019×10^{-11}

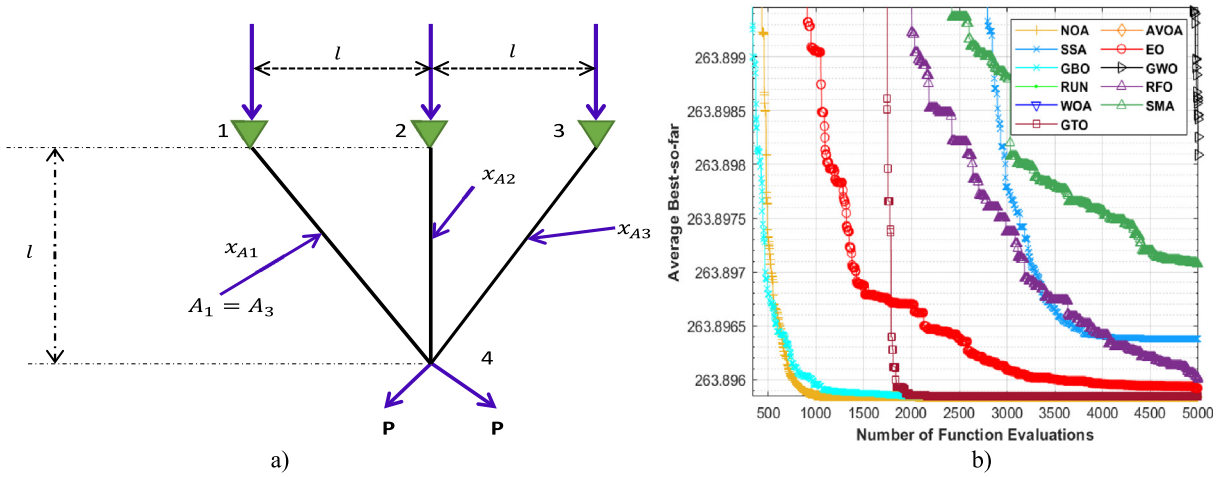


Fig. 22. Three-bar truss problem: (a) Shape and (b) Convergence curve.

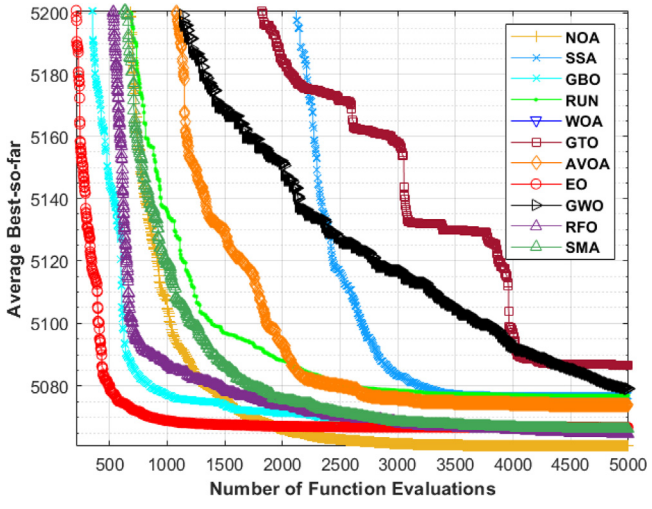


Fig. 23. Convergence curve of 10-bar truss design problem.

– Solutions of this problem are represented as follows:

Consider $X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} = [Ts \;\; Th \;\; R \;\; L]$

– The objective function employed along with the optimization algorithm to handle this problem is as:

$$\text{Minimize } f(X) = 0.6224x_1x_3x_4 + 1.7781x_3x_1^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$$

– Subject to the following constraints:

$$g_1(X) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(X) = -x_3 + 0.00954x_3 \le 0$$
$$g_3(X) = -\pi x_4x_3^2 - \frac{4}{3}x_3^3 + 1296000 \le 0$$
$$g_4(X) = x_4 - 240 \le 0$$

Variable range $0 \le x_1 \le 99$
$0 \le x_2 \le 99$
$10 \le x_3 \le 200$
$10 \le x_4 \le 200$

As shown in Table 15, NOA outperforms all other optimizers in terms of Ave, Worst, and Std values when solving the pressure vessel design problem. The averaged convergence curve obtained by the proposed algorithm when solving this problem is shown in Fig. 21(b) which affirms that NOA has a high convergence speed.

6.4. The three-bar truss design problem

Specifically, the goal of this problem is to determine the optimal value of three variables, which include the areas of bars 1, 2, and 3, to minimize the weight of the three-bar truss while also meeting certain constraints. Fig. 22 (a) depicts the general

Table 18
Description of some standard test functions.

ID	Functions	D	Domain	Global opt
F1	$\boldsymbol{F1} = \sum_{i=1}^{D} \boldsymbol{x_i^2}$	100	[−100,100]	0
F2	$\boldsymbol{F2} = \sum_{i=1}^{D} \lvert \boldsymbol{x_i} \rvert + \prod_{i=1}^{D} \lvert \boldsymbol{x_i} \rvert$	100	[−100,100]	0
F3	$\boldsymbol{F3} = \sum_{i=1}^{D} \left(\sum_{j=1}^{D} \boldsymbol{x_j} \right)^2$	100	[−100,100]	0
F4	$\boldsymbol{F4} = \max \left(\lvert \boldsymbol{x_i} \rvert \right) \lvert \boldsymbol{1} \le \boldsymbol{i} \le \boldsymbol{D}$	100	[−100,100]	0
F5	$\boldsymbol{F5} = \sum_{i=1}^{D-1} \left[100 \left(\boldsymbol{x_{i+1}} - \boldsymbol{x_i^2} \right)^2 + \left(\boldsymbol{x_i} - \boldsymbol{1} \right)^2 \right]$	100	[−30,30]	0
F6	$\boldsymbol{F6} = \sum_{i=1}^{D-1} [\boldsymbol{x_i} + \boldsymbol{0.5}]^2$	100	[−100,100]	0
F7	$\boldsymbol{F7} = \sum_{i=1}^{D} \boldsymbol{i x_i^4} + \boldsymbol{random}[\boldsymbol{0}, \boldsymbol{1}]$	100	[−1.28,1.28]	0
F8	$\boldsymbol{F8} = \sum_{i=1}^{D-1} -\boldsymbol{x_i} \boldsymbol{\sin} \left(\sqrt{\lvert \boldsymbol{x_i} \rvert} \right)$	100	[−500,500]	$-418.98 \times D$
F9	$\boldsymbol{F9} = \sum_{i=1}^{D} \left[\boldsymbol{x_i^2} - \boldsymbol{10} \cos \left(\boldsymbol{2\pi x_i} \right) + \boldsymbol{10} \right]$	100	[−5.12,5.12]	0
F10	$\boldsymbol{F10} = -\boldsymbol{20} \exp \left(-\boldsymbol{0.2} \sqrt{\frac{1}{\boldsymbol{D}} \sum_{i=1}^{D} \boldsymbol{x_i^2}} \right) - \exp \left(\frac{\boldsymbol{1}}{\boldsymbol{D}} \sum_{i=1}^{D} \cos \left(\boldsymbol{2\pi x_i} \right) \right) + \boldsymbol{20} + \boldsymbol{e}$	100	[−32,32]	0
F11	$\boldsymbol{F11} = \frac{\boldsymbol{1}}{\boldsymbol{4000}} \sum_{i=1}^{D} \boldsymbol{x_i^2} - \prod_{i=1}^{D} \cos \left(\frac{\boldsymbol{x_i}}{\sqrt{\boldsymbol{i}}} \right) + \boldsymbol{1}$	100	[−600,600]	0
F12	$\boldsymbol{F12} = \frac{\pi}{\boldsymbol{D}} \left(\boldsymbol{10} \sin \left(\pi \boldsymbol{y_1} \right) + \sum_{i=1}^{D-1} \left(\boldsymbol{y_i} - \boldsymbol{1} \right)^2 \left[\boldsymbol{1} + \boldsymbol{10} \sin^2 \left(\pi \boldsymbol{y_{i+1}} \right) \right] + \left(\boldsymbol{y_D} - \boldsymbol{1} \right)^2 \right) + \sum_{i=1}^{D} \boldsymbol{u}(\boldsymbol{x_i}, \boldsymbol{10}, \boldsymbol{100}, \boldsymbol{4})$	100	[−50,50]	0
F13	$\boldsymbol{F13} = \frac{\pi}{\boldsymbol{d}} \left(\boldsymbol{10} \sin^2 \left(\boldsymbol{3x_1} \right) + \sum_{i=1}^{D-1} \left(\boldsymbol{x_i} - \boldsymbol{1} \right)^2 \left[\boldsymbol{1} + \sin^2 \left(\boldsymbol{3\pi x_i} \right) \right] + \left(\boldsymbol{x_d} - \boldsymbol{1} \right)^2 \left[\boldsymbol{1} + \sin^2 \left(\boldsymbol{2\pi x_D} \right) \right] \right) + \sum_{i=1}^{d} \boldsymbol{u}(\boldsymbol{x_i}, \boldsymbol{5}, \boldsymbol{100}, \boldsymbol{4})$	100	[−50,50]	0
F14	$\boldsymbol{F14} = \left(\frac{\boldsymbol{1}}{\boldsymbol{500}} + \sum_{j=1}^{25} \frac{\boldsymbol{1}}{\boldsymbol{j} + \sum_{i=1}^{2} \left(\boldsymbol{x_i} - \boldsymbol{a_{ij}} \right)^2} \right)^{-1}$	2	[−65,65]	1
F15	$\boldsymbol{F15} = \sum_{i=1}^{11} \left[\boldsymbol{a_i} - \frac{\boldsymbol{x_1} \left(\boldsymbol{b_i^2} + \boldsymbol{b_i x_2} \right)}{\boldsymbol{b_i^2} + \boldsymbol{b_i x_3} + \boldsymbol{x_4}} \right]^2$	4	[−5,5]	0.00030
F16	$\boldsymbol{F16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
F17	$\boldsymbol{F17} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos \left(x_1 \right) + 10$	2	[−5,5]	0.398
F18	$\boldsymbol{F18} = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2 \right) \right] \times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2 \right) \right]$	2	[−2,2]	3
F19	$\boldsymbol{F19} = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij} \right)^2}$	3	[1,3]	−3.86
F20	$\boldsymbol{F20} = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{6} a_{ij} \left(x_j - p_{ij} \right)^2}$	6	[0,1]	−3.32

(*continued on next page*)

shape of the problem [106,107]. This problem is mathematically formulated as follows:

– Solutions to this problem are represented as follows:

$$X = \left[x_1 \ x_2 \right]$$

Table 18 (*continued*).

ID	Functions	D	Domain	Global opt
F21	$\boldsymbol{F21} = -\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
F22	$\boldsymbol{F22} = -\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.4028
F23	$\boldsymbol{F23} = -\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.5363

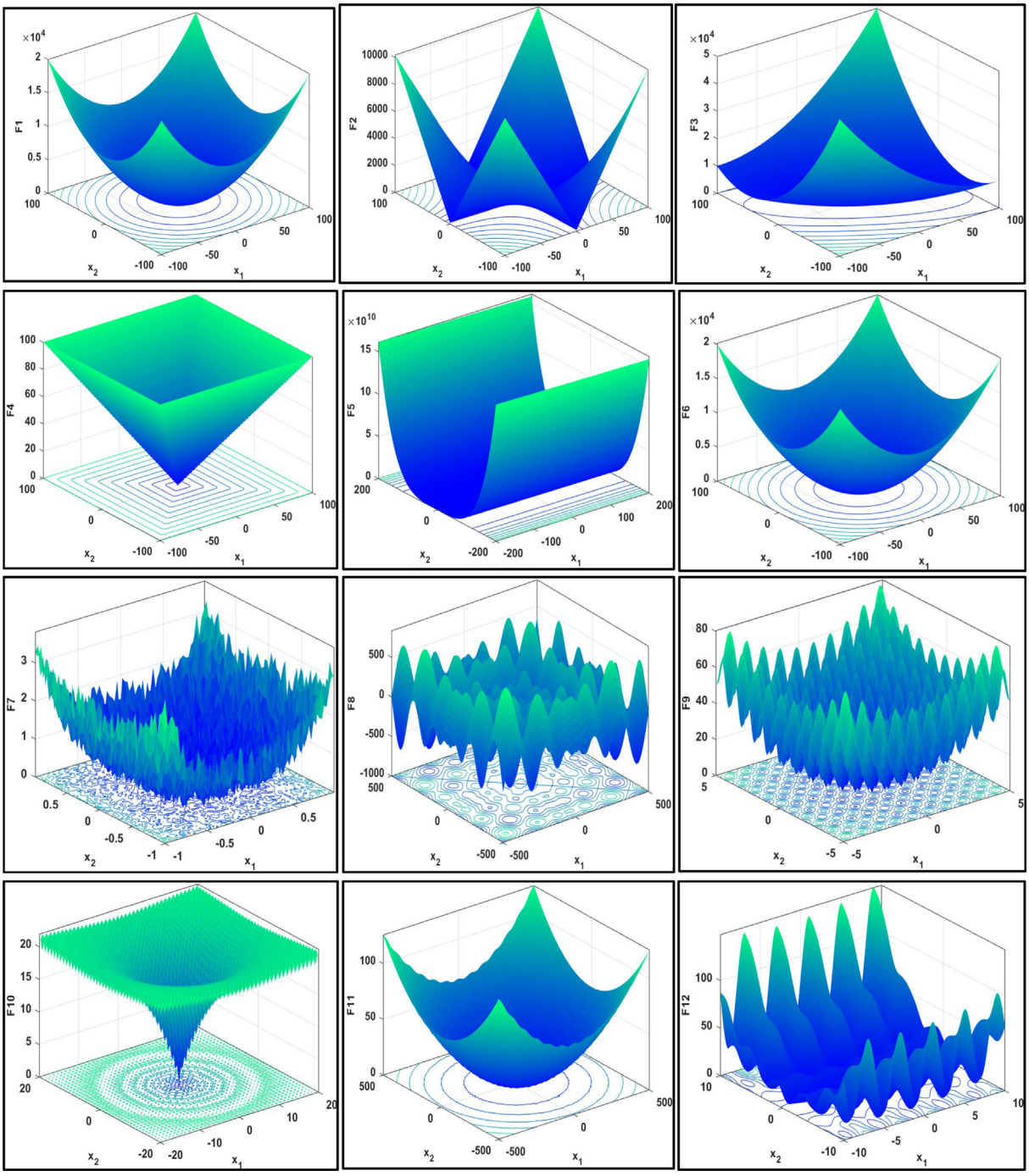


Fig. 24. Two-dimensional view of 23 well-known test functions.

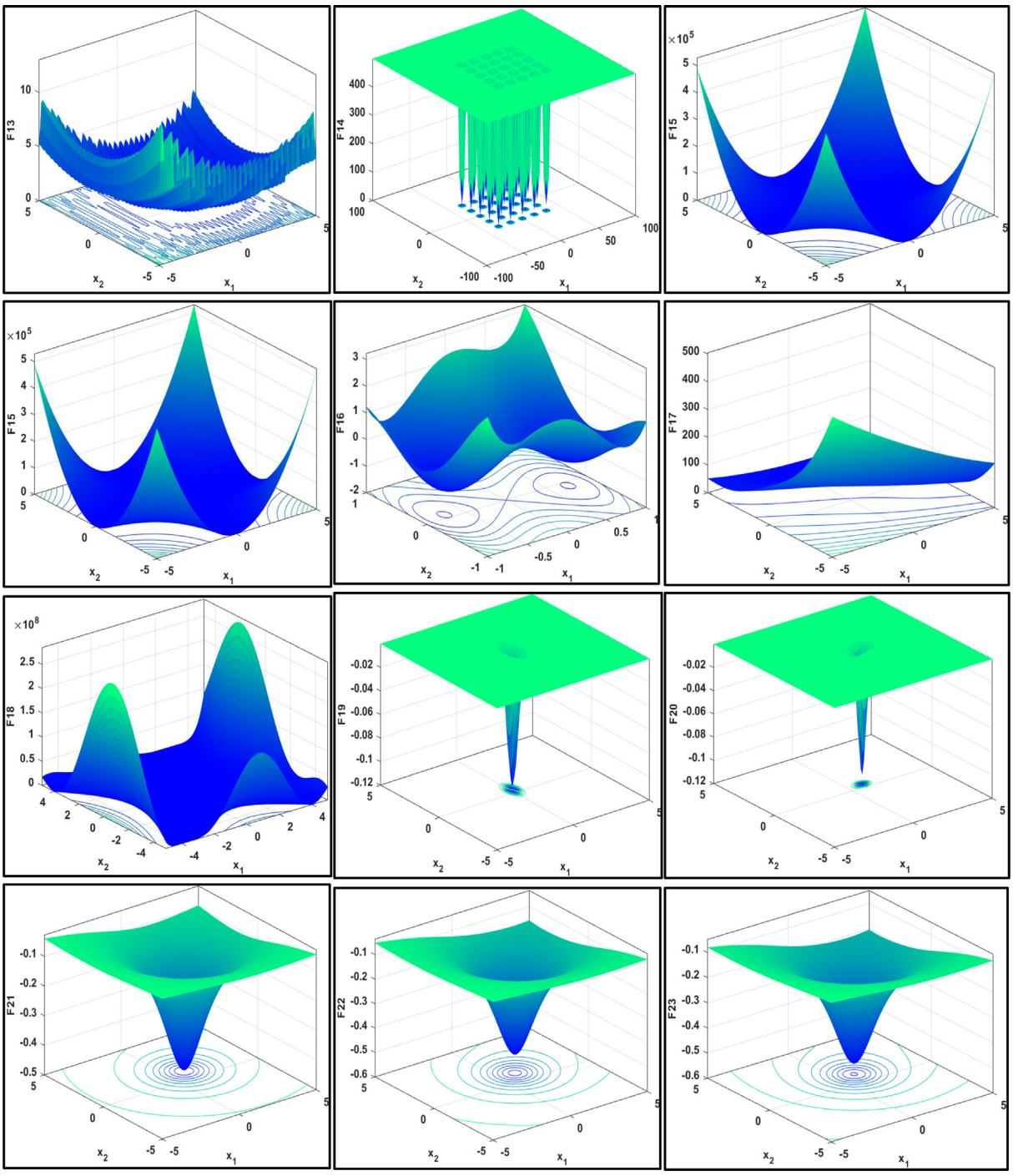


Fig. 24. (*continued*).

– The objective function employed along with the optimization algorithm to handle this problem is as:

$$\text{Minimize } f(X) = \left(2\sqrt{2}x_1 + x_2\right) \times l$$

– Subject to the following constraints:

$$g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_3(X) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0$$

Variable range $0 \leq x_1, x_2 \leq 1$

$$l = 100 \text{ cm}, P = 2\frac{KN}{\text{cm}^2}, \sigma = 2\frac{KN}{\text{cm}^2}$$

This problem is solved using NOA, and the results are compared to those obtained using the rival optimizers as shown in Table 16. From this table, NOA appears to be competitive with GBO and GTO in terms of the best, and Ave and superior to the others for the best, Ave, and Std of fitness values obtained from 30 independent runs. Table 16 also contains the optimal

Table 19
Characteristics of CEC-2014 test functions.

Type	ID	Functions	Global opt.	Domain
Unimodal function	F24 (CF1)	Function of Rotated High Conditioned Elliptic	100	[−100, 100]
	F25 (CF2)	Function of Rotated Bent Cigar	300	[−100, 100]
	F26 (CF3)	Function of Rotated Discus		
Simple multimodal Test functions	F27 (CF4)	Function of Shifted and Rotated Rosenbrock	400	[−100, 100]
	F28 (CF5)	Function of Shifted and Rotated Ackley	500	[−100, 100]
	F29 (CF6)	Function of Shifted and Rotated Weierstrass	600	[−100, 100]
	F30 (CF7)	Function of Shifted and Rotated Griewank	700	[−100, 100]
	F31 (CF8)	Function of Shifted Rastrigin	800	[−100, 100]
	F32 (CF9)	Function of Shifted and Rotated Rastrigin	900	[−100, 100]
	F33 (CF10)	Function of Shifted Schwefel	1000	[−100, 100]
	F34 (CF11)	Function of Shifted and Rotated Schwefel	1100	[−100, 100]
	F35 (CF12)	Function of Shifted and Rotated Katsuura	1200	[−100, 100]
	F36 (CF13)	Function of Shifted and Rotated HappyCat	1300	[−100, 100]
	F37 (CF14)	Function of Shifted and Rotated HGBat	1400	[−100, 100]
	F38 (CF15)	Function of Shifted and Rotated Expanded Griewank's plus Rosenbrock	1500	[−100, 100]
	F39 (CF16)	Function of Shifted and Rotated Expanded Scaffer	1600	[−100, 100]
Hybrid test functions	F40 (CF17)	Function 1 Hybrid	1700	[−100, 100]
	F41 (CF18)	Function 2 Hybrid	1800	[−100, 100]
	F42 (CF19)	Function 3 Hybrid	1900	[−100, 100]
	F43 (CF20)	Function 4 Hybrid	2000	[−100, 100]
	F44 (CF21)	Function 5 Hybrid	2100	[−100, 100]
	F45 (CF22)	Function 6 Hybrid	2200	[−100, 100]
Composition test functions	F46 (CF23)	Function 1 Composition	2300	[−100, 100]
	F47 (CF24)	Function 2 Composition	2400	[−100, 100]
	F48 (CF25)	Function 3 Composition	2500	[−100, 100]
	F49 (CF26)	Function 4 Composition	2600	[−100, 100]
	F50 (CF27)	Function 5 Composition	2700	[−100, 100]
	F51 (CF28)	Function 6 Composition	2800	[−100, 100]
	F52 (CF29)	Function 7 Composition	2900	[−100, 100]
	F53 (CF30)	Function 8 Composition	3000	[−100, 100]

Table 20
Characteristics of CEC-2017 test functions.

Type	ID	Functions	Global opt.	Domain
Unimodal function	F54 (CF1)	Function of Shifted and Rotated Bent Cigar	100	[−100, 100]
	F55 (CF3)	Function of Shifted and Rotated Zakharov	300	[−100, 100]
Simple multimodal Test functions	F56 (CF4)	Function of Shifted and Rotated Rosenbrock	400	[−100, 100]
	F57 (CF5)	Function of Shifted and Rotated Rastrigin	500	[−100, 100]
	F58 (CF6)	Function of Shifted and Rotated Expanded Scaffer	600	[−100, 100]
	F59 (CF7)	Function of Shifted and Rotated Lunacek Bi_Rastrigin	700	[−100, 100]
	F60 (CF8)	Function of Shifted and Rotated Non-Continuous Rastrigin	800	[−100, 100]
	F61 (CF9)	Function of Shifted and Rotated Levy	900	[−100, 100]
	F62 (CF10)	Function of Shifted and Rotated Schwefel	1000	[−100, 100]
Hybrid test functions	F63 (CF11)	Function 1 Hybrid	1100	[−100, 100]
	F64 (CF12)	Function 2 Hybrid	1200	[−100, 100]
	F65 (CF13)	Function 3 Hybrid	1300	[−100, 100]
	F66 (CF14)	Function 4 Hybrid	1400	[−100, 100]
	F67 (CF15)	Function 5 Hybrid	1500	[−100, 100]
	F68 (CF16)	Function 6 Hybrid	1600	[−100, 100]
	F69 (CF17)	Function 7 Hybrid	1700	[−100, 100]
	F70 (CF18)	Function 8 Hybrid	1800	[−100, 100]
	F71 (CF19)	Function 9 Hybrid	1900	[−100, 100]
	F72 (CF20)	Function 10 Hybrid	2000	[−100, 100]
Composition test functions	F73 (CF21)	Function 1 Composition	2100	[−100, 100]
	F74 (CF22)	Function 2 Composition	2200	[−100, 100]
	F75 (CF23)	Function 3 Composition	2300	[−100, 100]
	F76 (CF24)	Function 4 Composition	2400	[−100, 100]
	F77 (CF25)	Function 5 Composition	2500	[−100, 100]
	F78 (CF26)	Function 6 Composition	2600	[−100, 100]
	F79 (CF27)	Function 7 Composition	2700	[−100, 100]
	F80 (CF28)	Function 8 Composition	2800	[−100, 100]
	F81 (CF29)	Function 9 Composition	2900	[−100, 100]
	F82 (CF30)	Function 10 Composition	3000	[−100, 100]

values of the design variables, as well as the corresponding fitness values obtained by each algorithm. Fig. 22(b) depicts the averaged convergence curve obtained by the proposed algorithm when solving this problem, which demonstrates that NOA has a fast convergence speed.

6.5. The 10-bar truss design problem

The 10-bar truss design problem is the final engineering design problem that has been investigated in this study. This problem consists of ten design variables that must be precisely

Table 21
Characteristics of CEC-2020 test functions.

Type	ID	Functions	Global opt.	Domain
Unimodal	F83 (CF1)	Function of Shifted and Rotated Bent Cigar	100	[−100, 100]
Multimodal	F84 (CF2)	Function of Shifted and Rotated Lunacek Bi_Rastrigin	700	[−100, 100]
Hybrid test functions	F85 (CF3)	Function 1 Hybrid	1100	[−100, 100]
	F86 (CF4)	Function 2 Hybrid	1700	[−100, 100]
	F87 (CF5)	Function 3 Hybrid	1900	[−100, 100]
	F88 (CF6)	Function 4 Hybrid	2100	[−100, 100]
	F89 (CF7)	Function 5 Hybrid	1600	[−100, 100]
Hybrid test functions	F90 (CF8)	Function 1 Composition	2200	[−100, 100]
	F91 (CF9)	Function 2 Composition	2400	[−100, 100]
	F92 (CF10)	Function 3 Composition	2500	[−100, 100]

optimized to achieve the best possible design. This optimum design must be subjected to ten stress constraints and twelve displacement constraints to be considered complete. The mathematical model for this problem, as well as other relevant information, are available in [104].

This problem was studied using the NOA and rival algorithms, and the results are presented in Table 17. These results include the optimized design variables, as well as the best, Ave, Rank, and Std of the fitness values obtained by each algorithm throughout 30 separate runs. Examining this table reveals that NOA produces significantly better results than all of the other competing algorithms. The proposed algorithm's performance in terms of the maximum number of function evaluations required to reach the desired average fitness value must be evaluated, even though NOA may outperform the others in terms of the various statistical information under consideration. Because of this, Fig. 23 is presented to show the average fitness values obtained by the proposed algorithm during the optimization process, which proves that NOA has a rapid convergence rate.

7. Conclusion

This work proposes a novel MH algorithm known as the NOA as an alternative for solving optimization problems. The search, cache, and recovery behaviors of the nutcracker are the main inspiration of NOA. The performance of this algorithm was tested in a series of experiments. Twenty-three standard functions, test suites of CEC2014, CEC-2017, and CEC-2020, and five real-world engineering design problems were employed in this work to assess the proposed algorithm's exploratory capabilities and its exploitation, avoidance of local optima, and convergence. NOA was qualitatively evaluated using a variety of metrics, including search history, trajectory, the average fitness of solutions, and the best solution in each iteration. Scientifically, the Wilcoxon rank-sum test was used to assess the effectiveness of the proposed algorithm. The experimental results show that NOA can guarantee excellent exploration performance while maintaining an outstanding balance between exploration and exploitability, resulting in the superior performance of the proposed NOA. Five engineering problems (i.e., a welded beam, a tension/compression spring, a pressure vessel, a 3 bar truss, and a 10 bar truss) were solved in this study by obtaining additional evaluation. NOA outperformed all rival optimizers, including CEC2017 winners (LSHADE-SPACMA and LSHADE-cnEpSin), CEC2014 winners (L-SHADE and an adaptive variant of L-SHADE (AL-SHADE)), some of the highly-cited MH algorithms (WOA, SSA, and GWO), and some of the recently-published algorithms (SMA, GBO, EO, RUN, AVOA, RFO and GTO), in terms of the quality of outcomes and convergence speed. The NOA algorithm is currently being developed in binary and multi-objective variants under the names binary and multi-objective NOAs, respectively.

CRediT authorship contribution statement

Mohamed Abdel-Basset: Investigation, Methodology, Resources, Visualization, Software, Writing – original draft, Writing – review & editing. **Reda Mohamed:** Investigation, Methodology, Resources, Visualization, Software, Writing – original draft, Writing – review & editing. **Mohammed Jameel:** Conceptualization, Methodology, Writing – review & editing. **Mohamed Abouhawwash:** Conceptualization, Methodology, Resources, Visualization, Validation, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

We confirm that the manuscript has been read and approved by all named authors.

Appendix

See Tables 18–21 and Fig. 24.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, 1995.
[2] M. Dorigo, G. Di Caro, Ant colony optimization: A new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), IEEE, 1999.
[3] H.A. Abbass, MBO: Marriage in honey bees optimization-a haplometrosis polygynous swarming approach, in: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), IEEE, 2001.
[4] X. Li, A New Intelligent Optimization-Artificial Fish Swarm Algorithm (Doctor thesis), Zhejiang University of Zhejiang, China, 2003, p. 27.
[5] R. Martin, W. Stephen, Termite: A swarm intelligent routing algorithm for mobilewireless Ad-Hoc networks, in: Stigmergic Optimization, Springer, 2006, pp. 155–184.
[6] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[7] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing, NaBIC, Ieee, 2009.
[8] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspir. Comput. 2 (2) (2010) 78–84.
[9] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, Springer, 2010, pp. 65–74.

[10] X.-S. Yang, Flower pollination algorithm for global optimization, in: International Conference on Unconventional Computing and Natural Computation, Springer, 2012.
[11] A.H. Gandomi, A.H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.
[12] A. Askarzadeh, A. Rezazadeh, A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer, Int. J. Energy Res. 37 (10) (2013) 1196–1204.
[13] J.C. Bansal, et al., Spider monkey optimization algorithm for numerical optimization, Memetic Comput. 6 (1) (2014) 31–47.
[14] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[15] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[16] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[17] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[18] S. Mirjalili, et al., Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[19] S.H.S. Moosavi, V.K. Bardsiri, Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation, Eng. Appl. Artif. Intell. 60 (2017) 1–15.
[20] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50.
[21] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.
[22] A.A. Heidari, et al., Harris Hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[23] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowl.-Based Syst. 165 (2019) 169–196.
[24] S. Li, et al., Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[25] M. Khishe, M.R. Mosavi, Chimp optimization algorithm, Expert Syst. Appl. 149 (2020) 113338.
[26] D. Połap, M. Woźniak, Red fox optimization algorithm, Expert Syst. Appl. 166 (2021) 114107.
[27] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, Int. J. Intell. Syst. 36 (10) (2021) 5887–5958.
[28] L. Abualigah, et al., Aquila optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.
[29] K.M. Ong, P. Ong, C.K. Sia, A carnivorous plant algorithm for solving global optimization problems, Appl. Soft Comput. 98 (2021) 106833.
[30] W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, Comput. Methods Appl. Mech. Engrg. 388 (2022) 114194.
[31] S. Chamaani, S.A. Mirtaheri, M.S. Abrishamian, Improvement of time and frequency domain performance of antipodal Vivaldi antenna using multi-objective particle swarm optimization, IEEE Trans. Antennas and Propagation 59 (5) (2011) 1738–1742.
[32] L. Penghui, et al., Metaheuristic optimization algorithms hybridized with artificial intelligence model for soil temperature prediction: Novel model, IEEE Access 8 (2020) 51884–51904.
[33] S. Gao, et al., Ant colony optimization with clustering for solving the dynamic location routing problem, Appl. Math. Comput. 285 (2016) 149–173.
[34] S. Biswas, S. Acharyya, Neural model of gene regulatory network: A survey on supportive meta-heuristics, Theory Biosci. 135 (1) (2016) 1–19.
[35] P. Brown, et al., Fast and accurate non-sequential protein structure alignment using a new asymmetric linear sum assignment heuristic, Bioinformatics 32 (3) (2016) 370–377.
[36] N. Kumar, D. Kumar, An improved grey wolf optimization-based learning of artificial neural network for medical data classification, J. Inf. Commun. Technol. 20 (2) (2021) 213–248.
[37] P. Mohapatra, S. Chakravarty, P.K. Dash, An improved cuckoo search based extreme learning machine for medical data classification, Swarm Evol. Comput. 24 (2015) 25–49.
[38] G. Kalantzis, et al., Investigations of a GPU-based levy-firefly algorithm for constrained optimization of radiation therapy treatment planning, Swarm Evol. Comput. 26 (2016) 191–201.
[39] M.A. Mosa, Real-time data text mining based on gravitational search algorithm, Expert Syst. Appl. 137 (2019) 117–129.
[40] B. Gonzalez-Sanchez, M.A. Vega-Rodriguez, S. Santander-Jimenez, Multi-objective memetic meta-heuristic algorithm for encoding the same protein with multiple genes, Expert Syst. Appl. 136 (2019) 83–93.
[41] S. Li, W. Gong, Q. Gu, A comprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models, Renew. Sustain. Energy Rev. 141 (2021) 110828.
[42] B. Li, H. Chen, T. Tan, PV cell parameter extraction using data prediction–based meta-heuristic algorithm via extreme learning machine, Front. Energy Res. 9 (2021) 211.
[43] S. Li, et al., Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization, Energy Convers. Manag. 186 (2019) 293–305.
[44] W. Long, et al., A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models, Energy Convers. Manag. 203 (2020) 112243.
[45] D.S. AbdElminaam, et al., An Efficient Heap-Based Optimizer for Parameters Identification of Modified Photovoltaic Models, Ain Shams Eng. J. 13 (5) (2022) 101728.
[46] S. Mirghasemi, P. Andreae, M. Zhang, Domain-independent severely noisy image segmentation via adaptive wavelet shrinkage using particle swarm optimization and fuzzy C-means, Expert Syst. Appl. 133 (2019) 126–150.
[47] Q.-b. Zhang, P. Wang, Z. h. Chen, An improved particle filter for mobile robot localization based on particle swarm optimization, Expert Syst. Appl. 135 (2019) 181–193.
[48] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution–An updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[49] P.J. Van Laarhoven, E.H. Aarts, Simulated annealing, in: Simulated Annealing: Theory and Applications, Springer, 1987, pp. 7–15.
[50] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[51] R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[52] W. Kaidi, M. Khishe, M. Mohammadi, Dynamic Levy flight chimp optimization 235, 2022, p. 107625.
[53] S.-P. Gong, M. Khishe, M. Mohammadi, Niching chimp optimization for constraint multimodal engineering optimization problems, Expert Syst. Appl. 198 (2022) 116887.
[54] B. Wang, et al., Robust grey wolf optimizer for multimodal optimizations: A cross-dimensional coordination approach. 92 (3), 2022, pp. 1–30.
[55] A. Faramarzi, et al., Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[56] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[57] K.M. Dohms, T.M. Burg, Molecular markers reveal limited population genetic structure in a North American corvid, Clark's nutcracker (Nucifraga Columbiana), PLoS One 8 (11) (2013) e79621.
[58] P.A. Bednekoff, R.P. Balda, Clark's nutcracker spatial memory: The importance of large, structural cues, Behav. Process. 102 (2014) 12–17.
[59] H.E. Hutchins, R.M. Lanner, The central role of Clark's nutcracker in the dispersal and establishment of whitebark pine, Oecologia 55 (2) (1982) 192–201.
[60] D.F. Tomback, Foraging strategies of Clark's nutcracker, Living Bird 16 (1978) 123–161.
[61] D.F. Tomback, P. Achuff, Blister rust and western forest biodiversity: Ecology, values and outlook for white pines, Forest Pathol. 40 (3–4) (2010) 186–225.
[62] T. DF, Clark's Nutcracker (Nucifraga Columbiana). The Birds of North America Online, Cornell Lab of Ornithology, Ithaca, New York, 1998.
[63] S.T. McKinney, C.E. Fiedler, D.F. Tomback, Invasive pathogen threatens bird–pine mutualism: Implications for sustaining a high-elevation ecosystem, Ecol. Appl. 19 (3) (2009) 597–607.
[64] L.E. Barringer, et al., Whitebark pine stand condition, tree abundance, and cone production as predictors of visitation by Clark's nutcracker, PLoS One 7 (5) (2012) e37663.
[65] T.J. Lorenz, et al., Cache-site selection in Clark's Nutcracker (Nucifraga columbiana), Auk 128 (2) (2011) 237–247.
[66] S.B. Vander Wall, Dependence of Clark's nutcracker, Nucifraga Columbiana, on conifer seeds during the postfledging period, Can. Field Nat. 97 (1983) 208–214.
[67] S.B. Vander Wall, R.P. Balda, Coadaptations of the Clark's nutcracker and the pinon pine for efficient seed harvest and dispersal, Ecol. Monograph 47 (1) (1977) 89–111.
[68] S. Blackadder, Modelling whitebark pine distribution in the crown of the continent ecosystem, 2019, Arts.
[69] T. Bocsi, et al., Exploring the ecology of establishing oak trees in urban settings of the northeast, Cities Environ. (CATE) 14 (1) (2021) 3.
[70] C.R. Dimmick, Life History and the Development of Cache-Recovery Behaviors in Clark's Nutcracker, Northern Arizona University, 1993.
[71] T.D. Schaming, Clark's nutcracker breeding season space use and foraging behavior, PLoS One 11 (2) (2016) e0149116.
[72] M.E. Maier, Clark's Nutcracker Seed Harvest Patterns in Glacier National Park and a Novel Method for Monitoring Whitebark Pine Cones, Utah State University, 2012.

[73] D.F. Sherry, What food-storing birds remember, Can. J. Psychol./Rev. Can. de Psychol. 38 (2) (1984) 304.
[74] J.S. Pfadenhauer, F.A. Klötzli, Vegetation of the temperate high mountains, in: Global Vegetation, Springer., 2020, pp. 551–598.
[75] R. Gartshore, R. Broods, J. Somers, Limber pine seed harvest by Clark's nutcracker in the Sierra Nevada: Timing and foraging behavior, Condor 82 (4) (1980) 467–468.
[76] K.M. Christensen, T.G. Whitham, R.P. Balda, Discrimination among pinyon pine trees by Clark's nutcrackers: Effects of cone crop size and cone characters, Oecologia 86 (3) (1991) 402–407.
[77] S.B. Vander Wall, Foraging of Clark's nutcrackers on rapidly changing pine seed resources, Condor 90 (3) (1988) 621–631.
[78] D.F. Tomback, Dispersal of whitebark pine seeds by Clark's nutcracker: A mutualism hypothesis, J. Animal Ecol. (1982) 451–467.
[79] T.D. Schaming, Population-wide failure to breed in the Clark's nutcracker (Nucifraga columbiana), PLoS One 10 (5) (2015) e0123917.
[80] R.P. Balda, A. Kamil, The spatial memory of Clark's nutcrackers (Nucifraga columbiana) in an analogue of the radial arm maze, Pap. Beh. Biolog. Sci. (1988) 4.
[81] R.P. Balda, A.C. Kamil, Long-term spatial memory in Clark's nutcracker, Nucifraga Columbiana, Animal Behav. 44 (4) (1992) 761–769.
[82] B. Poucet, et al., The hippocampus and the neural code of spatial memory, Biol. Aujourd'hui 204 (2) (2010) 103–112.
[83] S.B. Vander Wall, An experimental analysis of cache recovery in Clark's nutcracker, Anim. Behav. 30 (1) (1982) 84–94.
[84] D.M. Kelly, et al., Effects of sun compass error on spatial search by Clark's nutcrackers, Integr. Zool. 14 (2) (2019) 172–181.
[85] D. Zhang, et al., Visual landmark-directed scatter-hoarding of Siberian chipmunks Tamias sibiricus, Integr. Zool. 11 (3) (2016) 175–181.
[86] S. Watanabe, Strategies of spatial learning for food storing in scrub jays, J. Ethol. 23 (2) (2005) 181–187.
[87] S. Li, et al., Slime mould algorithm: A new method for stochastic optimization. 111, 2020, pp. 300–323.
[88] M. Abdel-Basset, et al., Parameters identification of PV triple-diode model using improved generalized normal distribution algorithm, Mathematics 9 (9) (2021) 995.
[89] T.S. Ayyarao, P.P. Kumar, Parameter estimation of solar PV models with a new proposed war strategy optimization algorithm, Int. J. Energy Res. (2022).
[90] A.A.Z. Diab, et al., Coyote optimization algorithm for parameters estimation of various models of solar cells and PV modules, Ieee Access 8 (2020) 111102-111140.
[91] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Vol. 635, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013, p. 490.
[92] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017.
[93] J.J. Liang, et al., Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization, Computational Intelligence Laboratory, Zhengzhou University, 2019.
[94] I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, Inform. Sci. 540 (2020) 131–159.
[95] I. Ahmadianfar, et al., RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, Expert Syst. Appl. 181 (2021) 115079.
[96] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, Comput. Ind. Eng. 158 (2021) 107408.
[97] Y. Li, et al., A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem, 2022.
[98] R. Poláková, J. Tvrdík, P. Bujok, Evaluating the performance of L-SHADE with competing strategies on CEC2014 single parameter-operator test suite, in: 2016 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2016.
[99] A.W. Mohamed, et al., LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017.
[100] M.H. Nadimi-Shahraki, H. Zamani, DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization 198, 2022, p. 116895.
[101] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art, Comput. Methods Appl. Mech. Engrg. 191 (11–12) (2002) 1245–1287.
[102] Coello, C.A.C.J.C.m.i.a.m. and Engineering, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. 191(11-12), 2002, pp. 1245–1287.
[103] J. Arora, Introduction to Optimum Design, Elsevier, 2004.
[104] C.A.C. Coello, E.M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection 16 (3), 2002, pp. 193–203.
[105] A. Faramarzi, et al., Marine Predators Algorithm: A nature-inspired metaheuristic. 152, 2020, p. 113377.
[106] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: A new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[107] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.